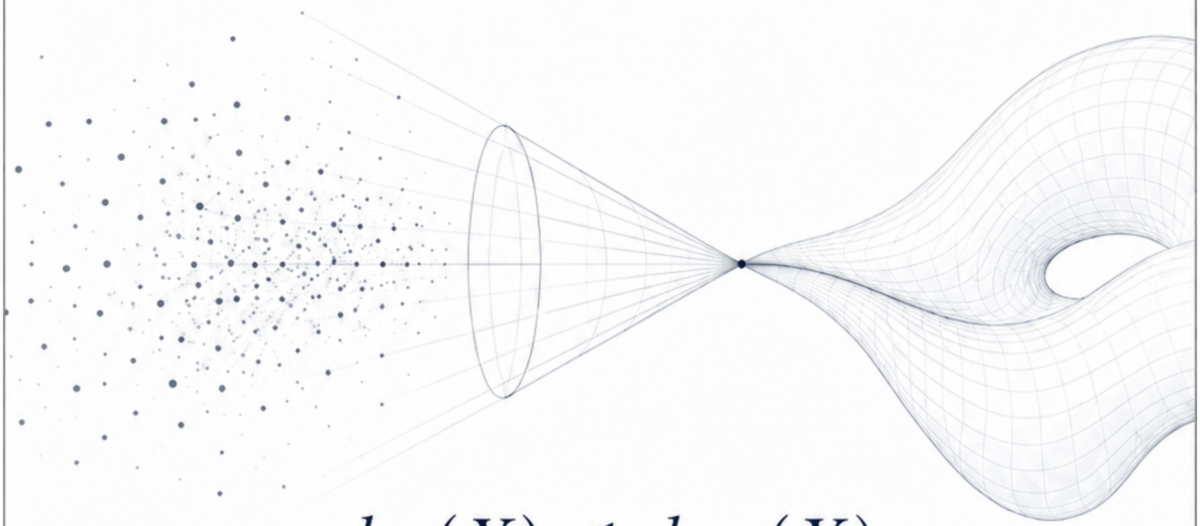
DRCC

Dimensional Reduction via Controlled Combinatorics



$$d_{\mathrm{rec}}(X) \leq d_{\mathrm{frag}}(X)$$

DRCC Criterion

DRCC is proposed as a toolbox for controlled reduction, designed to identify structural principles implicitly present in successful mathematical solution strategies.

DRCC-V2

Reza Hesamiy

Munich, Germany

2026

Notation and Symbols

This condensed reviewer manuscript uses the following notation.

Symbol	Meaning
$\mathcal{P}$	Class of admissible problem instances.
$P \in \mathcal{P}$	A finite problem instance.
$X(P)$	Classical candidate space associated with P.
$x \in X(P)$	A candidate solution, assignment, route, coloring, state, or decision option.
$X_{\mathrm{adm}}(P)$	Set of admissible candidates satisfying the constraints of P.
$d_{\mathrm{frag}}(P)$	Fragmentation dimension; $d_{\mathrm{frag}}(P) = \lvert X(P)\rvert$.
C_P	Controlled reduction map $C_P : X(P) \to Z(P)$.
$Z(P)$	Reduced structural state space associated with P.
$\zeta \in Z(P)$	A reduced structural state or class.
$C_P^{-1}(\zeta)$	Reduction class of all candidates mapped to ζ.
$\Omega_P(\zeta)$	Reconstruction space associated with the reduced state ζ.
$d_{\mathrm{rec}}(P,\zeta)$	Reconstruction dimension; $d_{\mathrm{rec}}(P,\zeta) = \lvert\Omega_P(\zeta)\rvert$.
R	Controlled reconstruction operator.
$N_{\mathrm{Classic}}(P)$	Classical evaluation count.
$N_{\mathrm{Collapse}}(P)$	Reduction or collapse evaluation count.
$N_{\mathrm{Reconstruction}}(P,\zeta)$	Reconstruction evaluation count.
$N_{\mathrm{DRCC}}(P,\zeta)$	Total DRCC evaluation count.
$T_{\mathrm{Classic}}(P)$	Classical runtime.
$T_{\mathrm{DRCC}}(P,\zeta)$	DRCC runtime.
f	Reference operation rate used to convert counts into runtime.
$G(P,\zeta)$	Runtime gain of DRCC relative to classical enumeration.
$W = (n_w, R_w[s])$	Discrete transition point.
$\Delta_N(n)$	Count-level gap $\Delta_N(n) = N_{\mathrm{Classic}}(n) - N_{\mathrm{DRCC}}(n)$.
$\Delta_T(n)$	Runtime gap $\Delta_T(n) = T_{\mathrm{Classic}}(n) - T_{\mathrm{DRCC}}(n)$.

All symbols are used in a finite combinatorial sense unless explicitly stated otherwise.
In particular, the reconstruction space $\Omega_P(\zeta)$ is always a finite set in this condensed reviewer manuscript.

Abstract

This condensed reviewer manuscript presents a runtime-oriented formulation of Dimensional Reduction via Controlled Combinatorics (DRCC-V2).

DRCC is a finite, discrete framework for studying reconstruction problems through controlled structural reduction. The guiding principle is to preserve the information required for admissible reconstruction and to collapse distinctions that are irrelevant to the reconstruction task.

The manuscript defines the classical candidate space $X(P)$, the fragmentation dimension

$$d_{\text{frag}}(P) = |X(P)|,$$

the controlled reduction map

$$C_P : X(P) \to Z(P),$$

the reconstruction space

$$\Omega_P(\zeta) = \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\},$$

and the reconstruction dimension

$$d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|.$$

The operational stability condition is

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty.$$

The runtime model compares classical enumeration with a DRCC decomposition into controlled reduction and reconstruction:

$$T_{\text{DRCC}} = T_{\text{Collapse}} + T_{\text{Reconstruction}}.$$

The runtime gain is defined as

$$G(P, \zeta) = \frac{T_{\text{Classic}}(P)}{T_{\text{DRCC}}(P, \zeta)}.$$

The transition point

$$W = (n_w, R_w[s])$$

marks the discrete size at which the classical and DRCC runtime models coincide.

The manuscript applies this framework to finite case studies including housing selection, travelling-salesman-type routing, constraint satisfaction problems, Boolean satisfiability, graph 3-coloring, and the full adder.

These examples are used to illustrate how controlled reduction, reconstruction dimension, runtime gain, and transition behavior can be computed or bounded in concrete settings.

The work does not claim a proof of $P = NP$ or $P \neq NP$. It does not claim a universal speedup for graph coloring, satisfiability, constraint satisfaction, or routing problems. Its contribution is a compact mathematical language and finite runtime accounting framework for identifying when controlled structural reduction may become advantageous relative to classical enumeration.

Contents

1 Core Definitions and Runtime Framework

1.1 Purpose of the Condensed Manuscript

This condensed manuscript presents the runtime and structural reconstruction core of Dimensional Reduction via Controlled Combinatorics.

It is designed as a reviewer-oriented version of DRCC-V2.

The objective is not to claim universal computational improvement. The objective is to provide a finite mathematical framework for analyzing when controlled structural reduction and admissible reconstruction may become advantageous relative to classical enumeration.

The manuscript focuses on seven elements:

1. finite candidate spaces;
2. controlled reduction maps;
3. admissible reconstruction spaces;
4. fragmentation and reconstruction dimensions;
5. runtime-count decomposition;
6. transition behavior;
7. explicit non-claims.

All statements are interpreted in a finite combinatorial setting.

1.2 Problem Instances and Candidate Spaces

Definition 1.1 (Problem Space)**.** *Let*

$$\mathcal{P} \tag{1.1}$$

denote a class of admissible finite problem instances.
An element

$$P \in \mathcal{P} \tag{1.2}$$

is called a problem instance.

Examples of problem instances considered in this manuscript include housing selection problems, travelling-salesman-type routing problems, constraint satisfaction problems, Boolean satisfiability instances, graph coloring instances, and structured digital systems.

Definition 1.2 (Candidate Space)**.** *For a problem instance $P \in \mathcal{P}$, the candidate space is a finite set*

$$X(P) = \{x_1, x_2, \ldots, x_N\}. \tag{1.3}$$

Each element

$$x \in X(P) \tag{1.4}$$

is a candidate solution, assignment, route, coloring, configuration, or decision option associated with P.

In the classical exhaustive model, solving P requires evaluating all elements of $X(P)$, unless additional structure is used.

Definition 1.3 (Fragmentation Dimension). *The fragmentation dimension of P is defined by*

$$d_{\text{frag}}(P) = |X(P)|. \tag{1.5}$$

The quantity $d_{\text{frag}}(P)$ is used operationally as the size of the pre-reduction candidate space.

1.3 Controlled Reduction

Definition 1.4 (Controlled Reduction Map). *Let $P \in \mathcal{P}$ be a finite problem instance with candidate space $X(P)$.*
A controlled reduction map is a mapping

$$C_P : X(P) \to Z(P), \tag{1.6}$$

where $Z(P)$ is the reduced structural state space associated with P.
For a candidate $x \in X(P)$, the value

$$\zeta = C_P(x) \tag{1.7}$$

is called a reduced state or structural class.

The reduction map C_P is not required to be injective.
Indeed, the purpose of DRCC is often to identify candidates that are distinct in the classical candidate space but equivalent with respect to a chosen reconstruction task.

Definition 1.5 (Reduction Class). *For $\zeta \in Z(P)$, the reduction class associated with ζ is*

$$C_P^{-1}(\zeta) = \{x \in X(P) : C_P(x) = \zeta\}. \tag{1.8}$$

A reduction class contains all candidates that collapse to the same reduced structural state.

1.4 Reconstruction Spaces

Definition 1.6 (Admissible Candidate Space). *Let*

$$X_{\text{adm}}(P) \subseteq X(P) \tag{1.9}$$

denote the set of candidates that satisfy the admissibility conditions of the problem instance P.

The admissible candidate space may be equal to the full candidate space, or it may be a proper subset determined by constraints.

Definition 1.7 (Reconstruction Space). *Let $P \in \mathcal{P}$ be a finite problem instance and let $\zeta \in Z(P)$ be a reduced state.*
The reconstruction space associated with ζ is defined by

$$\Omega_P(\zeta) = \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}. \tag{1.10}$$

The reconstruction space contains precisely those admissible candidates that are compatible with the reduced state ζ.

Definition 1.8 (Reconstruction Dimension)**.** *The reconstruction dimension associated with* ζ *is defined by*

$$d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|. \tag{1.11}$$

Thus, $d_{\text{rec}}(P, \zeta)$ measures the size of the admissible reconstruction space after controlled reduction.

1.5 Structural Stability Condition

Definition 1.9 (DRCC Structural Stability)**.** *A reduced state* $\zeta \in Z(P)$ *is called structurally stable if*

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty. \tag{1.12}$$

This condition excludes three failure modes.
First,

$$d_{\text{rec}}(P, \zeta) = 0 \tag{1.13}$$

means that no admissible reconstruction exists.
Second,

$$d_{\text{rec}}(P, \zeta) > d_{\text{frag}}(P) \tag{1.14}$$

would contradict the interpretation of reconstruction as a controlled substructure of the original candidate space.
Third,

$$d_{\text{frag}}(P) = \infty \tag{1.15}$$

would move the analysis outside the finite combinatorial setting of this manuscript.

Remark 1.10 (Operational Nature of the Stability Condition)**.** *The condition*

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty \tag{1.16}$$

does not claim that every reduced problem is easy.
It states only that reconstruction remains finite, non-empty, and structurally bounded by the original candidate space.

1.6 Classical Runtime Model

Definition 1.11 (Classical Evaluation Count)**.** *Let* $P \in \mathcal{P}$ *be a finite problem instance.*
The classical evaluation count is

$$N_{\text{Classic}}(P) = |X(P)| \cdot c_{\text{eval}}(P), \tag{1.17}$$

where $c_{\text{eval}}(P) \geq 1$ *denotes the cost of evaluating one candidate in the classical model.*

In the unit-cost model,

$$c_{\text{eval}}(P) = 1, \tag{1.18}$$

and therefore

$$N_{\text{Classic}}(P) = d_{\text{frag}}(P). \tag{1.19}$$

Definition 1.12 (Classical Runtime)**.** *Let $f > 0$ denote the reference operation rate. The classical runtime is*

$$T_{\text{Classic}}(P) = \frac{N_{\text{Classic}}(P)}{f}. \tag{1.20}$$

The reference rate f converts operation counts into wall-clock time. The structural comparison is governed by the evaluation count.

1.7 DRCC Runtime Model

The DRCC runtime is decomposed into a reduction phase and a reconstruction phase.

Definition 1.13 (Collapse Evaluation Count)**.** *The collapse evaluation count is denoted by*

$$N_{\text{Collapse}}(P). \tag{1.21}$$

It measures the number of operations required to construct reduced states, structural classes, fragments, or candidate groups.

Definition 1.14 (Reconstruction Evaluation Count)**.** *For a reduced state ζ, the reconstruction evaluation count is*

$$N_{\text{Reconstruction}}(P, \zeta) = d_{\text{rec}}(P, \zeta) c_{\text{rec}}(P, \zeta), \tag{1.22}$$

where $c_{\text{rec}}(P, \zeta) \geq 1$ denotes the cost of evaluating or constructing one admissible reconstruction.

Definition 1.15 (DRCC Evaluation Count)**.** *The DRCC evaluation count is*

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta). \tag{1.23}$$

Definition 1.16 (DRCC Runtime)**.** *The DRCC runtime is*

$$T_{\text{DRCC}}(P, \zeta) = \frac{N_{\text{DRCC}}(P, \zeta)}{f}. \tag{1.24}$$

Equivalently,

$$T_{\text{DRCC}}(P, \zeta) = T_{\text{Collapse}}(P) + T_{\text{Reconstruction}}(P, \zeta), \tag{1.25}$$

where

$$T_{\text{Collapse}}(P) = \frac{N_{\text{Collapse}}(P)}{f} \tag{1.26}$$

and

$$T_{\text{Reconstruction}}(P, \zeta) = \frac{N_{\text{Reconstruction}}(P, \zeta)}{f}. \tag{1.27}$$

1.8 Runtime Gain

Definition 1.17 (Runtime Gain)**.** *The runtime gain of DRCC relative to classical enumeration is*

$$G(P,\zeta) = \frac{T_{\text{Classic}}(P)}{T_{\text{DRCC}}(P,\zeta)}. \tag{1.28}$$

Since both runtimes use the same reference rate f, this is equivalently

$$G(P,\zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P,\zeta)}. \tag{1.29}$$

The interpretation is

$$G(P,\zeta) < 1 \quad \Longrightarrow \quad \text{classical enumeration is smaller,} \tag{1.30}$$

$$G(P,\zeta) = 1 \quad \Longrightarrow \quad \text{transition point,} \tag{1.31}$$

and

$$G(P,\zeta) > 1 \quad \Longrightarrow \quad \text{DRCC is smaller under the model.} \tag{1.32}$$

Remark 1.18 (No Universal Speedup Claim)**.** *The condition*

$$G(P,\zeta) > 1 \tag{1.33}$$

is model-dependent.

It does not imply that DRCC is universally faster for all instances of a problem class.

It states only that, for the specified instance, reduction map, reconstruction space, and cost model, the DRCC count is smaller than the corresponding classical count.

1.9 Transition Point

For small instances, the overhead of reduction may dominate.

For larger or more structured instances, controlled reduction may become advantageous.

Definition 1.19 (Discrete Transition Point)**.** *Let $n \in \mathbb{N}$ denote a discrete problem-size parameter.*

A discrete transition point is a pair

$$W = (n_w, R_w[s]), \qquad n_w \in \mathbb{N}, \tag{1.34}$$

such that

$$T_{\text{Classic}}(n_w) = T_{\text{DRCC}}(n_w) = R_w[s]. \tag{1.35}$$

At the transition point,

$$G(n_w) = 1. \tag{1.36}$$

The transition point is not universal. It depends on the problem family, the reduction map, the reconstruction method, and the cost model.

1.10 Runtime Gap

Definition 1.20 (Runtime Gap)**.** *The runtime gap is defined by*

$$\Delta_T(n) = T_{\text{Classic}}(n) - T_{\text{DRCC}}(n). \tag{1.37}$$

At the count level, the corresponding gap is

$$\Delta_N(n) = N_{\text{Classic}}(n) - N_{\text{DRCC}}(n). \tag{1.38}$$

The sign of $\Delta_N(n)$ determines the count-level regime:

$$\Delta_N(n) < 0 \quad \Longrightarrow \quad \text{classical enumeration is smaller,} \tag{1.39}$$

$$\Delta_N(n) = 0 \quad \Longrightarrow \quad \text{transition,} \tag{1.40}$$

and

$$\Delta_N(n) > 0 \quad \Longrightarrow \quad \text{DRCC is smaller under the model.} \tag{1.41}$$

1.11 Operational Runtime Protocol

For every case study in this manuscript, the same finite runtime protocol is used:

1. define the classical candidate space $X(P)$;
2. compute or estimate $d_{\text{frag}}(P) = |X(P)|$;
3. define the controlled reduction map C_P;
4. determine the reduced state ζ;
5. construct or estimate $\Omega_P(\zeta)$;
6. compute or estimate $d_{\text{rec}}(P, \zeta)$;
7. compute $N_{\text{Classic}}(P)$;
8. compute $N_{\text{DRCC}}(P, \zeta)$;
9. compute $G(P, \zeta)$;
10. identify $W = (n_w, R_w[s])$, when applicable.

This protocol makes the case studies comparable.

1.12 Summary

The central quantities introduced in this chapter are

$$d_{\text{frag}}(P) = |X(P)|, \tag{1.42}$$

$$d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|, \tag{1.43}$$

$$0 < d_{\text{rec}}(P,\zeta) \le d_{\text{frag}}(P) < \infty, \tag{1.44}$$

$$N_{\text{DRCC}}(P,\zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P,\zeta), \tag{1.45}$$

$$G(P,\zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P,\zeta)}, \tag{1.46}$$

and

$$W = (n_w, R_w[s]). \tag{1.47}$$

These definitions provide the common mathematical language for the case studies developed in the following chapters.

2 Algorithmic Protocol and Empirical Setup

2.1 Purpose of the Chapter

The purpose of this chapter is to make the DRCC runtime framework operational.
The previous chapter introduced the core quantities

$$d_{\text{frag}}, \qquad d_{\text{rec}}, \qquad N_{\text{Classic}}, \qquad N_{\text{DRCC}}, \qquad G, \qquad W. \tag{2.1}$$

This chapter explains how these quantities are computed or estimated in a reproducible finite setting.
The objective is not to present an optimized software implementation.
The objective is to define a concrete protocol that can be applied consistently across all case studies in this manuscript.
The protocol separates three layers:

1. classical enumeration;
2. controlled DRCC reduction and reconstruction;
3. runtime comparison and transition detection.

This separation is important because DRCC is compared against a specified classical baseline and a specified reconstruction procedure.

2.2 Input Data

Let

$$P \in \mathcal{P} \tag{2.2}$$

be a finite problem instance.
The algorithmic protocol assumes the following input data:

$$P, \qquad X(P), \qquad C_P, \qquad A_{\text{adm}}, \qquad f. \tag{2.3}$$

Here, P is the problem instance, $X(P)$ is the classical candidate space, C_P is the controlled reduction map, A_{adm} is the admissibility rule set, and f is the reference evaluation rate used to convert counts into wall-clock time.

The reference rate f is measured in evaluations per second.

Unless otherwise stated, count-level comparisons are independent of the particular choice of f, because

$$G(P,\zeta) = \frac{N_{\mathrm{Classic}}(P)}{N_{\mathrm{DRCC}}(P,\zeta)}. \tag{2.4}$$

2.3 Output Data

The protocol returns the following structural quantities:

$$d_{\mathrm{frag}}(P), \qquad \zeta, \qquad \Omega_P(\zeta), \qquad d_{\mathrm{rec}}(P,\zeta). \tag{2.5}$$

It also returns the corresponding count and runtime quantities:

$$N_{\mathrm{Classic}}(P), \qquad N_{\mathrm{DRCC}}(P,\zeta), \qquad T_{\mathrm{Classic}}(P), \qquad T_{\mathrm{DRCC}}(P,\zeta). \tag{2.6}$$

Finally, it reports

$$G(P,\zeta), \qquad W = (n_w, R_w[s]), \tag{2.7}$$

when a parameterized family of instances is available.

For a single instance, the protocol reports the runtime gain but does not infer a transition point.

2.4 Classical Enumeration Baseline

The classical baseline evaluates every candidate in the candidate space.

The input data are

$$P, \qquad X(P), \qquad A_{\mathrm{adm}}, \qquad f. \tag{2.8}$$

The output data are

$$N_{\mathrm{Classic}}(P), \qquad T_{\mathrm{Classic}}(P). \tag{2.9}$$

The protocol is:

1. construct or define the candidate space $X(P)$;
2. compute $d_{\mathrm{frag}}(P) = |X(P)|$;
3. evaluate each candidate $x \in X(P)$;
4. count the number of evaluations;
5. convert the count into runtime.

The classical count is

$$N_{\mathrm{Classic}}(P) = |X(P)| c_{\mathrm{eval}}(P). \tag{2.10}$$

The corresponding runtime is

$$T_{\text{Classic}}(P) = \frac{N_{\text{Classic}}(P)}{f}. \tag{2.11}$$

In the unit-cost model,

$$c_{\text{eval}}(P) = 1, \tag{2.12}$$

and therefore

$$N_{\text{Classic}}(P) = d_{\text{frag}}(P). \tag{2.13}$$

2.5 DRCC Reduction and Reconstruction

The DRCC protocol first reduces the candidate space and then reconstructs admissible candidates from the reduced representation.
The input data are

$$P, \qquad X(P), \qquad C_P, \qquad A_{\text{adm}}, \qquad f. \tag{2.14}$$

The output data are

$$\zeta, \qquad \Omega_P(\zeta), \qquad N_{\text{DRCC}}(P,\zeta), \qquad T_{\text{DRCC}}(P,\zeta). \tag{2.15}$$

The controlled reduction map is

$$C_P : X(P) \to Z(P). \tag{2.16}$$

For a reduced state

$$\zeta \in Z(P), \tag{2.17}$$

the reconstruction space is

$$\Omega_P(\zeta) = \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}. \tag{2.18}$$

The reconstruction dimension is

$$d_{\text{rec}}(P,\zeta) = |\Omega_P(\zeta)|. \tag{2.19}$$

The collapse count is

$$N_{\text{Collapse}}(P). \tag{2.20}$$

The reconstruction count is

$$N_{\text{Reconstruction}}(P,\zeta) = d_{\text{rec}}(P,\zeta)c_{\text{rec}}(P,\zeta). \tag{2.21}$$

The total DRCC count is

$$N_{\text{DRCC}}(P,\zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P,\zeta). \tag{2.22}$$

The DRCC runtime is

$$T_{\mathrm{DRCC}}(P,\zeta) = \frac{N_{\mathrm{DRCC}}(P,\zeta)}{f}. \tag{2.23}$$

The reduced state is accepted only if

$$0 < d_{\mathrm{rec}}(P,\zeta) \le d_{\mathrm{frag}}(P) < \infty. \tag{2.24}$$

If this admissibility condition fails, the reduced state is not accepted as a stable DRCC reconstruction state.

2.6 Runtime Gain and Transition Detection

For a parameterized family of problem instances

$$(P_n)_{n\in\mathbb{N}}, \tag{2.25}$$

the runtime gain is computed at each problem size n.
For each n, compute

$$T_{\mathrm{Classic}}(n) \tag{2.26}$$

and

$$T_{\mathrm{DRCC}}(n). \tag{2.27}$$

The runtime gap is

$$\Delta_T(n) = T_{\mathrm{Classic}}(n) - T_{\mathrm{DRCC}}(n). \tag{2.28}$$

The runtime gain is

$$G(n) = \frac{T_{\mathrm{Classic}}(n)}{T_{\mathrm{DRCC}}(n)}. \tag{2.29}$$

The first transition index n_w satisfies

$$G(n_w) \ge 1 \tag{2.30}$$

with at least one smaller tested value $n < n_w$ satisfying

$$G(n) < 1. \tag{2.31}$$

If exact equality occurs, the transition point is

$$W = (n_w, R_w[s]), \tag{2.32}$$

where

$$R_w[s] = T_{\mathrm{Classic}}(n_w) = T_{\mathrm{DRCC}}(n_w). \tag{2.33}$$

If exact equality does not occur, the transition may be reported as a crossing interval

$$W_{\mathrm{int}} = [n_-, n_+], \tag{2.34}$$

where

$$G(n_-) < 1 \qquad \text{and} \qquad G(n_+) > 1. \tag{2.35}$$

2.7 Count-Level Validation

The first validation layer is count-level validation.

This level does not depend on implementation language, processor speed, memory hierarchy, compiler optimization, or machine-specific execution details.

The reported quantities are

$$|X(P)|, \qquad |\Omega_P(\zeta)|, \qquad N_{\text{Collapse}}, \qquad N_{\text{Reconstruction}}, \qquad N_{\text{DRCC}}. \tag{2.36}$$

The count-level runtime gain is

$$G_{\text{count}}(P,\zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P,\zeta)}. \tag{2.37}$$

This is the primary validation level used in the case studies.

It is transparent and reproducible.

2.8 Wall-Clock Validation

The second validation layer is wall-clock validation.

Here one measures actual runtime on a concrete machine.

For wall-clock validation, the following data should be reported.

Field	**Required Information**
Processor	CPU or hardware platform used for the experiment.
Memory	Available memory and relevant memory limits.
Programming language	Implementation language and version.
Instance generator	How the problem instances were generated.
Classical implementation	Description of the baseline enumeration.
DRCC implementation	Description of the reduction and reconstruction procedure.
Measured quantities	Runtime, candidate counts, reconstruction counts, and gain.
Repetition protocol	Number of runs and averaging method.

Wall-clock validation is useful, but it must be interpreted with care.

A poorly optimized implementation may hide structural gain.

Conversely, an optimized implementation may overstate improvement if the classical baseline is not implemented fairly.

Therefore, wall-clock experiments should be accompanied by count-level validation.

2.9 Fair Comparison Conditions

A runtime comparison between classical enumeration and DRCC is accepted only if the following fairness conditions are satisfied:

F1. Both methods are applied to the same problem instance P.

F2. Both methods aim at the same admissibility or solution criterion.

F3. The classical candidate space $X(P)$ is explicitly defined.

F4. The DRCC reduction map C_P is explicitly defined.

F5. The reconstruction space $\Omega_P(\zeta)$ is explicitly defined or constructively computable.

F6. No hidden oracle is used.

F7. The collapse cost N_{Collapse} is included.

F8. The parameters and instance-generation procedure are stated.

These conditions are included to prevent overinterpretation of runtime gain.

2.10 Empirical Evidence Template

Each case study may use a table of the following form.

Instance	N_{Classic}	N_{Collapse}	N_{Rec}	N_{DRCC}	G
P_1	–	–	–	–	–
P_2	–	–	–	–	–
P_3	–	–	–	–	–

The placeholders are replaced in the case-study chapters by instance-specific counts.

2.11 Minimal Pseudocode

The core DRCC runtime procedure can be written in compact pseudocode.

Input: $P, X(P), C_P, A_{\text{adm}}, f$.
Output: $T_{\text{Classic}}, T_{\text{DRCC}}, G, W$.

1. $d_{\text{frag}}(P) \leftarrow |X(P)|$.
2. $N_{\text{Classic}}(P) \leftarrow d_{\text{frag}}(P) c_{\text{eval}}(P)$.
3. $\zeta \leftarrow C_P(X(P))$.
4. $\Omega_P(\zeta) \leftarrow \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}$.
5. $d_{\text{rec}}(P,\zeta) \leftarrow |\Omega_P(\zeta)|$.
6. $N_{\text{DRCC}} \leftarrow N_{\text{Collapse}} + d_{\text{rec}}(P,\zeta) c_{\text{rec}}(P,\zeta)$.
7. $T_{\text{Classic}} \leftarrow N_{\text{Classic}}/f$.
8. $T_{\text{DRCC}} \leftarrow N_{\text{DRCC}}/f$.
9. $G \leftarrow T_{\text{Classic}}/T_{\text{DRCC}}$.
10. Accept only if $0 < d_{\text{rec}}(P,\zeta) \le d_{\text{frag}}(P) < \infty$.

This pseudocode is intentionally abstract.
Each case study specifies how $X(P)$, C_P, and $\Omega_P(\zeta)$ are instantiated.

2.12 Interpretation

The algorithmic protocol clarifies the meaning of empirical evidence in this manuscript.
DRCC is not evaluated by intuition alone.
It is evaluated by

$$N_{\text{Classic}}, \qquad N_{\text{DRCC}}, \qquad G, \qquad W. \tag{2.38}$$

The key comparison is between two counted procedures:

$$\text{classical enumeration} \tag{2.39}$$

and

$$\text{controlled reduction plus admissible reconstruction.} \tag{2.40}$$

This is the operational basis for the case studies developed in the following chapters.

2.13 Summary

This chapter introduced the algorithmic protocol used throughout the condensed manuscript.

The protocol defines a classical enumeration baseline, a DRCC reduction and reconstruction procedure, a runtime-gain calculation, a transition detection method, validation layers, and fairness conditions.

Each case study reports the same core quantities:

$$d_{\text{frag}}, \qquad d_{\text{rec}}, \qquad N_{\text{Classic}}, \qquad N_{\text{DRCC}}, \qquad G, \qquad W. \tag{2.41}$$

3 Housing Problem

3.1 Problem Definition

The housing problem is a finite multi-criteria selection problem.

A finite set of apartments is given by

$$H = \{h_1, h_2, \dots, h_n\}. \tag{3.1}$$

Each apartment is described by a finite list of attributes such as budget, location, size, public transport access, energy efficiency, parking, and availability.

Let

$$m \in \mathbb{N} \tag{3.2}$$

denote the number of structural criteria used for the first selection stage.

The task is to identify apartments compatible with the requirements and then select one or more admissible candidates.

This case is used as a reviewer-oriented runtime example because it has a transparent candidate space, a natural filtering structure, and a measurable reduction from all available apartments to a smaller candidate set.

3.2 Classical Runtime Model

In the classical baseline, every apartment is evaluated against all criteria.

The classical candidate space is

$$X_{\text{Housing}}(P) = H. \tag{3.3}$$

Thus,

$$d_{\text{frag}}(P) = |H| = n. \tag{3.4}$$

Let

$$c_{\text{crit}} \tag{3.5}$$

denote the cost of checking one criterion for one apartment.

The structural screening cost per apartment is

$$c_{\text{screen}} = m c_{\text{crit}}. \tag{3.6}$$

If the classical procedure also performs detailed inspection for every apartment, let

$$c_{\text{detail}} \tag{3.7}$$

denote the cost of detailed inspection per apartment.

The classical evaluation count is

$$N_{\text{Classic}} = n(c_{\text{screen}} + c_{\text{detail}}). \tag{3.8}$$

The corresponding runtime is

$$T_{\text{Classic}} = \frac{N_{\text{Classic}}}{f}. \tag{3.9}$$

Remark 3.1. *If*

$$c_{\text{detail}} = 0, \tag{3.10}$$

then the housing problem is only a basic filtering task.

In that case, DRCC does not automatically provide a runtime advantage.

A runtime advantage appears only when the reduction prevents expensive detail operations from being applied to all apartments.

3.3 DRCC Reduction Model

The DRCC reduction model separates the housing task into two phases.

The first phase performs structural filtering.

The second phase performs detailed reconstruction or selection only inside the reduced candidate set.

Let

$$C_{\text{Housing}} : H \to Z_{\text{Housing}} \tag{3.11}$$

be a reduction map that assigns each apartment to a structural compatibility state.

For example, an apartment may be classified according to budget, location, energy condition, availability, and accessibility.

Let

$$\zeta \in Z_{\text{Housing}} \tag{3.12}$$

denote the reduced state corresponding to apartments that pass the structural filters.

The reduced candidate set is

$$C_\zeta = \{h \in H : C_{\text{Housing}}(h) = \zeta\}. \tag{3.13}$$

Let

$$|C_\zeta| = c \tag{3.14}$$

be the number of remaining candidates after structural reduction.
In the housing example, c is expected to be much smaller than n when the criteria are selective.

3.4 Candidate Collapse

The candidate collapse is the finite reduction

$$n \longrightarrow c. \tag{3.15}$$

The collapse ratio is

$$\rho_{\text{Housing}} = \frac{n}{c}. \tag{3.16}$$

If

$$\rho_{\text{Housing}} > 1, \tag{3.17}$$

then the reduction is nontrivial.
However, a nontrivial reduction alone does not imply runtime gain.
Runtime gain depends on the collapse cost, reconstruction cost, and the cost avoided by not applying detailed inspection to all apartments.

3.5 Reconstruction Space

The admissible reconstruction space for the housing problem is

$$\Omega_{\text{Housing}}(\zeta) = \{h \in H_{\text{adm}} : C_{\text{Housing}}(h) = \zeta\}. \tag{3.18}$$

Here,

$$H_{\text{adm}} \subseteq H \tag{3.19}$$

denotes the set of apartments satisfying the admissibility requirements.
The reconstruction dimension is

$$d_{\text{rec}}(P, \zeta) = |\Omega_{\text{Housing}}(\zeta)|. \tag{3.20}$$

In the simplest case,

$$d_{\text{rec}}(P, \zeta) = c. \tag{3.21}$$

The structural stability condition is

$$0 < d_{\text{rec}}(P, \zeta) \le d_{\text{frag}}(P) < \infty. \tag{3.22}$$

Substituting the housing quantities gives

$$0 < c \le n < \infty. \tag{3.23}$$

3.6 DRCC Runtime Model

The DRCC evaluation count is decomposed into collapse cost and reconstruction cost.
The collapse cost is

$$N_{\text{Collapse}} = nc_{\text{screen}}. \tag{3.24}$$

The reconstruction cost is

$$N_{\text{Reconstruction}} = cc_{\text{detail}}. \tag{3.25}$$

Therefore,

$$N_{\text{DRCC}} = nc_{\text{screen}} + cc_{\text{detail}}. \tag{3.26}$$

The DRCC runtime is

$$T_{\text{DRCC}} = \frac{N_{\text{DRCC}}}{f}. \tag{3.27}$$

3.7 Runtime Comparison

The classical count is

$$N_{\text{Classic}} = n(c_{\text{screen}} + c_{\text{detail}}). \tag{3.28}$$

The DRCC count is

$$N_{\text{DRCC}} = nc_{\text{screen}} + cc_{\text{detail}}. \tag{3.29}$$

The count-level gap is

$$\Delta_N = N_{\text{Classic}} - N_{\text{DRCC}}. \tag{3.30}$$

Substituting the two counts gives

$$\Delta_N = n(c_{\text{screen}} + c_{\text{detail}}) - (nc_{\text{screen}} + cc_{\text{detail}}). \tag{3.31}$$

Hence,

$$\Delta_N = (n - c)c_{\text{detail}}. \tag{3.32}$$

Therefore,

$$\Delta_N > 0 \iff n > c \text{ and } c_{\text{detail}} > 0. \tag{3.33}$$

This equation expresses the main runtime principle of the housing case.
DRCC becomes advantageous only when the reduced candidate set is smaller than the original set and when detailed evaluation has nonzero cost.

3.8 Runtime Gain

The runtime gain is

$$G_{\text{Housing}} = \frac{N_{\text{Classic}}}{N_{\text{DRCC}}}. \tag{3.34}$$

Using the preceding equations, one obtains

$$G_{\text{Housing}} = \frac{n(c_{\text{screen}} + c_{\text{detail}})}{nc_{\text{screen}} + cc_{\text{detail}}}. \tag{3.35}$$

If

$$c_{\text{detail}} \gg c_{\text{screen}} \tag{3.36}$$

and

$$c \ll n, \tag{3.37}$$

then G_{Housing} can be significantly larger than one.
If

$$c_{\text{detail}} = 0, \tag{3.38}$$

then

$$G_{\text{Housing}} = 1. \tag{3.39}$$

Thus, DRCC does not claim automatic speedup for ordinary filtering.

The gain appears when structural reduction avoids expensive detailed inspection of candidates already eliminated by admissibility constraints.

3.9 Numerical Count-Level Example

Consider the following count-level example:

$$n = 1200, \qquad m = 7, \qquad c = 25. \tag{3.40}$$

Assume unit criterion cost

$$c_{\text{crit}} = 1. \tag{3.41}$$

Then

$$c_{\text{screen}} = mc_{\text{crit}} = 7. \tag{3.42}$$

Assume detailed inspection costs

$$c_{\text{detail}} = 1000 \tag{3.43}$$

operation units per apartment.
The classical count is

$$N_{\text{Classic}} = 1200(7 + 1000) = 1{,}208{,}400. \tag{3.44}$$

The DRCC count is

$$N_{\text{DRCC}} = 1200 \cdot 7 + 25 \cdot 1000 = 33{,}400. \tag{3.45}$$

The runtime gain is

$$G_{\text{Housing}} = \frac{1{,}208{,}400}{33{,}400} \approx 36.18. \tag{3.46}$$

This number should not be interpreted as a universal housing speedup.
It is a count-level result under the stated cost model.

3.10 Evidence Table

Quantity	**Value**	**Interpretation**
n	1200	Number of available apartments.
m	7	Number of structural criteria.
c	25	Candidates after controlled reduction.
d_{frag}	1200	Initial candidate size.
d_{rec}	25	Reconstruction candidate size.
N_{Classic}	1,208,400	Classical count under the stated model.
N_{DRCC}	33,400	DRCC count under the stated model.
G	≈ 36.18	Count-level runtime gain.

3.11 Transition Point

For the housing model, suppose that the reduced candidate count is approximately constant:

$$c(n) = c_0. \tag{3.47}$$

The classical count is

$$N_{\text{Classic}}(n) = n(c_{\text{screen}} + c_{\text{detail}}), \tag{3.48}$$

and the DRCC count is

$$N_{\text{DRCC}}(n) = nc_{\text{screen}} + c_0 c_{\text{detail}}. \tag{3.49}$$

The transition condition is

$$N_{\text{Classic}}(n_w) = N_{\text{DRCC}}(n_w). \tag{3.50}$$

Substituting the two count models gives

$$n_w(c_{\text{screen}} + c_{\text{detail}}) = n_w c_{\text{screen}} + c_0 c_{\text{detail}}. \tag{3.51}$$

If

$$c_{\text{detail}} > 0, \tag{3.52}$$

then

$$n_w = c_0. \tag{3.53}$$

Thus, in the idealized constant-candidate model, DRCC becomes advantageous once the number of available apartments exceeds the expected reduced candidate count.

If an additional fixed overhead

$$h > 0 \tag{3.54}$$

is introduced, then

$$N_{\text{DRCC}}(n) = h + nc_{\text{screen}} + c_0 c_{\text{detail}}. \tag{3.55}$$

The transition point becomes

$$n_w = c_0 + \frac{h}{c_{\text{detail}}}. \tag{3.56}$$

This expression makes explicit that overhead delays the transition.

3.12 Finite Runtime Interpretation

For the housing model, the comparison is entirely finite and count-level.

The classical count is

$$N_{\text{Classic}}(n) = n(c_{\text{screen}} + c_{\text{detail}}). \tag{3.57}$$

The DRCC count is

$$N_{\text{DRCC}}(n) = nc_{\text{screen}} + c_0 c_{\text{detail}}. \tag{3.58}$$

The count-level gap is

$$\Delta_N(n) = N_{\text{Classic}}(n) - N_{\text{DRCC}}(n). \tag{3.59}$$

Thus,

$$\Delta_N(n) = (n - c_0)c_{\text{detail}}. \tag{3.60}$$

The transition occurs when

$$n = c_0. \tag{3.61}$$

The interpretation is finite and operational:

DRCC becomes favorable when the number of available candidates exceeds the reduced candidate count and when detailed evaluation has nonzero cost.

3.13 Interpretation

The housing case shows a practical form of DRCC.

The classical method evaluates every apartment in detail.

The DRCC method first applies structural filters, then performs detailed inspection only inside the reduced reconstruction space.

The key condition for runtime gain is

$$c < n \tag{3.62}$$

together with

$$c_{\text{detail}} > 0. \tag{3.63}$$

Thus, the housing example supports the following limited claim:

$$\boxed{\text{For multi-stage selection problems with expensive detailed evaluation, controlled reduction can reduce runtime by restricting detailed inspection to an admissible reconstruction space.}} \tag{3.64}$$

It does not support the stronger claim that DRCC automatically improves all filtering problems.

3.14 Summary

For the housing problem, the core quantities are

$$d_{\text{frag}} = n, \qquad d_{\text{rec}} = c. \tag{3.65}$$

The structural stability condition is

$$0 < c \leq n < \infty. \tag{3.66}$$

The runtime models are

$$N_{\text{Classic}} = n(c_{\text{screen}} + c_{\text{detail}}) \tag{3.67}$$

and

$$N_{\text{DRCC}} = nc_{\text{screen}} + cc_{\text{detail}}. \tag{3.68}$$

The runtime gain is

$$G_{\text{Housing}} = \frac{n(c_{\text{screen}} + c_{\text{detail}})}{nc_{\text{screen}} + cc_{\text{detail}}}. \tag{3.69}$$

The transition point in the constant-candidate model is

$$n_w = c. \tag{3.70}$$

The housing case therefore provides a clear example of DRCC as controlled candidate collapse followed by admissible reconstruction.

4 TSG / TSP Routing Problem

4.1 Problem Definition

Let

$$G = (V, E, w) \tag{4.1}$$

be a complete weighted graph with

$$|V| = n. \tag{4.2}$$

The vertices represent locations, and the edge-weight function

$$w : E \to \mathbb{R}_{\geq 0} \tag{4.3}$$

assigns a non-negative travel cost to every edge.
A tour is a Hamiltonian cycle visiting every vertex exactly once and returning to the starting vertex.
Let

$$\mathcal{T}_n \tag{4.4}$$

denote the set of symmetric tours, where tours are identified up to cyclic rotation and reversal.
The objective of the classical travelling-salesman-type problem is to find

$$\tau^* = \arg\min_{\tau \in \mathcal{T}_n} w(\tau), \tag{4.5}$$

where

$$w(\tau) = \sum_{e \in \tau} w(e). \tag{4.6}$$

In this manuscript, the TSG / TSP example is used to study runtime growth and structural reduction.
It is not used to claim a general solution of the TSP.

4.2 Classical Tour Enumeration

For the symmetric TSP on n labeled vertices, the number of distinct Hamiltonian cycles is

$$|\mathcal{T}_n| = \frac{(n-1)!}{2}. \tag{4.7}$$

Thus, the classical candidate space is

$$X_{\mathrm{TSP}}(P) = \mathcal{T}_n, \tag{4.8}$$

and the fragmentation dimension is

$$d_{\mathrm{frag}}(P) = |X_{\mathrm{TSP}}(P)| = |\mathcal{T}_n| = \frac{(n-1)!}{2}. \tag{4.9}$$

Under the unit-cost model, the classical evaluation count is

$$N_{\text{Classic}}(n) = \frac{(n-1)!}{2}. \tag{4.10}$$

The corresponding runtime at reference evaluation rate f is

$$T_{\text{Classic}}(n) = \frac{(n-1)!}{2f}. \tag{4.11}$$

This factorial growth is the main reason why exhaustive enumeration quickly becomes infeasible.

4.3 DRCC Structural Fragment

DRCC replaces direct enumeration of all tours by a structural reduction step followed by controlled reconstruction.

Let

$$F_G = (V, E_F) \tag{4.12}$$

be a sparse structural fragment of the graph.

A typical choice is a connected reference graph such as a minimum spanning tree or another admissible structural skeleton.

For the present runtime model, assume that

$$|E_F| = n - 1. \tag{4.13}$$

The fragment F_G does not itself solve the TSP.

It supplies a structural reference against which tours are classified.

4.4 DRCC Structural Classes

For a tour

$$\tau \in \mathcal{T}_n, \tag{4.14}$$

define its fragment signature by

$$C_F(\tau) = E(\tau) \cap E_F. \tag{4.15}$$

Thus, $C_F(\tau)$ records which fragment edges occur in the tour.

The controlled reduction map is

$$C_F : \mathcal{T}_n \to Z_F, \tag{4.16}$$

where

$$Z_F \subseteq \mathcal{P}(E_F) \tag{4.17}$$

is the set of feasible fragment signatures.

Since

$$|E_F| = n - 1, \tag{4.18}$$

one has the upper bound

$$|Z_F| \le 2^{n-1}. \tag{4.19}$$

The reduction is therefore

$$\frac{(n-1)!}{2} \longrightarrow |Z_F| \le 2^{n-1}. \tag{4.20}$$

This does not by itself solve the optimization problem.
It partitions the tour space into structural reconstruction classes.

4.5 Reconstruction Spaces

For a fragment signature

$$\zeta \in Z_F, \tag{4.21}$$

the corresponding reconstruction space is

$$\Omega_F(\zeta) = \{\tau \in \mathcal{T}_n : C_F(\tau) = \zeta\}. \tag{4.22}$$

The reconstruction dimension is

$$d_{\text{rec}}(P, \zeta) = |\Omega_F(\zeta)|. \tag{4.23}$$

The structural stability condition becomes

$$0 < |\Omega_F(\zeta)| \le |\mathcal{T}_n| < \infty. \tag{4.24}$$

This condition states that a signature is admissible only if it admits at least one tour and remains a finite subspace of the full tour space.

4.6 Controlled Reconstruction of Tours

A reconstruction procedure assigns to each feasible signature ζ one or more candidate tours.
Let

$$R_F(\zeta) \subseteq \Omega_F(\zeta) \tag{4.25}$$

be the set of tours reconstructed from ζ.
Let

$$r(\zeta) = |R_F(\zeta)| \tag{4.26}$$

denote the reconstruction budget for the class ζ.
The total reconstruction count is

$$N_{\text{Reconstruction}} = \sum_{\zeta \in Z_F} r(\zeta). \tag{4.27}$$

If

$$r(\zeta) \le r_{\max} \quad \text{for every feasible signature } \zeta, \tag{4.28}$$

then

$$N_{\text{Reconstruction}} \le |Z_F| r_{\max} \le 2^{n-1} r_{\max}. \tag{4.29}$$

4.7 Objective-Preserving Reconstruction

For optimization problems, a reduction must preserve enough information to recover an optimal solution.

This motivates the following condition.

Definition 4.1 (Objective-Preserving Reconstruction)**.** *A reconstruction procedure is called objective-preserving if, for every non-empty reconstruction space $\Omega_F(\zeta)$, it returns a tour*

$$\tau_\zeta \in \Omega_F(\zeta) \tag{4.30}$$

satisfying

$$w(\tau_\zeta) = \min_{\tau \in \Omega_F(\zeta)} w(\tau). \tag{4.31}$$

Proposition 4.2 (Classwise Optimality Implies Global Optimality)**.** *Assume that the feasible reconstruction spaces $\Omega_F(\zeta)$ partition $\mathcal{T}_n$.*

If, for every non-empty class $\Omega_F(\zeta)$, the reconstruction procedure returns a class-optimal representative τ_ζ, then

$$\min_{\zeta \in Z_F} w(\tau_\zeta) = \min_{\tau \in \mathcal{T}_n} w(\tau). \tag{4.32}$$

Proof. Since the reconstruction spaces partition $\mathcal{T}_n$, every tour belongs to exactly one non-empty class $\Omega_F(\zeta)$.

For each such class, τ_ζ is assumed to minimize the tour weight inside that class.

Therefore, minimizing over all class representatives is equivalent to minimizing over all tours in the union of the classes.

Since this union is $\mathcal{T}_n$, the equality follows. □

Remark 4.3. *This proposition does not state that classwise optimal reconstruction is always easy.*

It states only that if classwise optimal reconstruction is achieved, then global optimality is preserved.

In the runtime model, the cost of classwise reconstruction must be counted explicitly.

4.8 DRCC Runtime Model

Let

$$N_{\text{Fragment}}(n) \tag{4.33}$$

denote the cost of constructing the structural fragment F_G.

For a dense graph, a simple count-level model is

$$N_{\text{Fragment}}(n) = n^2. \tag{4.34}$$

The DRCC count is

$$N_{\text{DRCC}}(n) = N_{\text{Fragment}}(n) + N_{\text{Reconstruction}}(n). \tag{4.35}$$

Using the bounded reconstruction model,

$$N_{\text{DRCC}}(n) \leq n^2 + 2^{n-1} r_{\text{max}}. \tag{4.36}$$

The corresponding runtime is

$$T_{\text{DRCC}}(n) = \frac{N_{\text{DRCC}}(n)}{f}. \tag{4.37}$$

4.9 Runtime Gain

The runtime gain is

$$G_{\text{TSP}}(n) = \frac{N_{\text{Classic}}(n)}{N_{\text{DRCC}}(n)}. \tag{4.38}$$

Using the classical count and the bounded DRCC count gives the estimate

$$G_{\text{TSP}}(n) \geq \frac{\frac{(n-1)!}{2}}{n^2 + 2^{n-1} r_{\text{max}}}. \tag{4.39}$$

This expression shows the structural runtime mechanism.

The classical numerator grows factorially.

The DRCC denominator grows according to fragment construction and controlled reconstruction.

If r_{max} remains bounded or grows slowly, then the DRCC count grows much more slowly than classical enumeration.

4.10 Count-Level Example

The following example uses

$$r_{\text{max}} = 3 \tag{4.40}$$

as an illustrative reconstruction budget.

The DRCC count model is

$$N_{\text{DRCC}}(n) = n^2 + 3 \cdot 2^{n-1}. \tag{4.41}$$

The classical count is

$$N_{\text{Classic}}(n) = \frac{(n-1)!}{2}. \tag{4.42}$$

n	N_{Classic}	N_{DRCC}	G
6	60	132	0.45
8	2520	448	5.63
10	181440	1636	110.90
12	19958400	6288	3173.41

This table is not a claim of universal TSP speedup.

It illustrates the count-level effect of replacing factorial tour enumeration by structural classes and bounded reconstruction.

4.11 Transition Point

In the illustrative model, the transition occurs when

$$\frac{(n-1)!}{2} = n^2 + 3 \cdot 2^{n-1}. \tag{4.43}$$

The table shows that

$$G(6) < 1 \tag{4.44}$$

and

$$G(8) > 1. \tag{4.45}$$

Thus the discrete crossing interval is

$$W_{\text{int}} = [6, 8]. \tag{4.46}$$

In this model, DRCC overhead dominates at $n = 6$, but structural reduction becomes advantageous by $n = 8$.

4.12 Factorial Enumeration and Finite DRCC Bound

The classical TSP enumeration count grows factorially:

$$N_{\text{Classic}}(n) = \frac{(n-1)!}{2}. \tag{4.47}$$

The DRCC structural-signature model satisfies

$$|Z_F| \leq 2^{n-1}. \tag{4.48}$$

With bounded reconstruction budget $r_{\max}$, the DRCC count satisfies

$$N_{\text{DRCC}}(n) \leq n^2 + 2^{n-1} r_{\max}. \tag{4.49}$$

Thus, the finite comparison is between

$$\frac{(n-1)!}{2} \tag{4.50}$$

and

$$n^2 + 2^{n-1} r_{\max}. \tag{4.51}$$

This is a count-level comparison.

It does not claim a general polynomial-time solution of the TSP.

It only states that, under the stated structural-signature model and bounded reconstruction budget, the DRCC count may grow more slowly than full tour enumeration.

4.13 Interpretation

The TSG / TSP case demonstrates a strong form of runtime contrast in this condensed manuscript.

The classical method enumerates tours.

DRCC classifies tours by structural signatures and reconstructs admissible tours inside reconstruction spaces.

The essential reduction is

$$\mathcal{T}_n \longrightarrow Z_F \longrightarrow \Omega_F(\zeta). \tag{4.52}$$

The runtime advantage depends on three conditions:

1. the number of structural classes must be much smaller than the number of tours;
2. the reconstruction budget per class must remain controlled;
3. if optimality is claimed, reconstruction must preserve classwise optima.

Without the third condition, DRCC yields admissible reconstructed tours but not necessarily the globally optimal TSP tour.

4.14 Summary

For the TSG / TSP case, the classical candidate count is

$$N_{\text{Classic}}(n) = \frac{(n-1)!}{2}. \tag{4.53}$$

A DRCC structural-signature model satisfies

$$|Z_F| \leq 2^{n-1}. \tag{4.54}$$

With bounded reconstruction budget $r_{\max}$,

$$N_{\text{DRCC}}(n) \leq n^2 + 2^{n-1} r_{\max}. \tag{4.55}$$

The runtime gain is

$$G_{\text{TSP}}(n) = \frac{N_{\text{Classic}}(n)}{N_{\text{DRCC}}(n)}. \tag{4.56}$$

In the illustrative example with $r_{\max} = 3$, the transition interval is

$$W_{\text{int}} = [6, 8]. \tag{4.57}$$

The TSG / TSP case therefore provides a reviewer-relevant example of how DRCC can convert factorial enumeration into a controlled reconstruction model, under explicit structural assumptions and without claiming a general solution of the TSP.

5 Constraint Satisfaction Problems

5.1 Problem Definition

A finite constraint satisfaction problem is a tuple

$$P_{\text{CSP}} = (V, D, \mathcal{C}). \tag{5.1}$$

Here

$$V = \{x_1, x_2, \ldots, x_n\} \tag{5.2}$$

is a finite set of variables.
Each variable x_i has a finite domain

$$D_i. \tag{5.3}$$

For simplicity, assume in the uniform-domain case that

$$|D_i| = q \qquad \text{for all } i = 1, \ldots, n. \tag{5.4}$$

The set

$$\mathcal{C} = \{C_1, C_2, \ldots, C_m\} \tag{5.5}$$

is a finite set of constraints.
Each constraint C_j acts on a scope

$$S_j \subseteq V. \tag{5.6}$$

The arity of the constraint is

$$|S_j|. \tag{5.7}$$

Let

$$r = \max_{1 \le j \le m} |S_j| \tag{5.8}$$

be the maximum constraint arity.
The objective is to find an assignment

$$a : V \to \bigcup_{i=1}^{n} D_i \tag{5.9}$$

such that

$$a(x_i) \in D_i \qquad \text{for all } i = 1, \ldots, n, \tag{5.10}$$

and every constraint in $\mathcal{C}$ is satisfied.

5.2 Classical Search Space

The classical candidate space consists of all complete assignments.
In the uniform-domain case,

$$X_{\mathrm{CSP}}(P) = D_1 \times D_2 \times \cdots \times D_n. \tag{5.11}$$

Therefore,

$$d_{\mathrm{frag}}(P) = |X_{\mathrm{CSP}}(P)| = q^n. \tag{5.12}$$

If each complete assignment is checked against all m constraints, then the classical evaluation count is

$$N_{\mathrm{Classic}}(P) = mq^n. \tag{5.13}$$

The corresponding runtime is

$$T_{\mathrm{Classic}}(P) = \frac{mq^n}{f}. \tag{5.14}$$

This is the exponential baseline for the CSP case study.

5.3 Constraint-Induced Collapse

DRCC does not begin by enumerating all complete assignments.
Instead, it first constructs local admissible fragments induced by the constraints.
For each constraint C_j, define the local admissible table

$$A_j = \left\{ u \in \prod_{x_i \in S_j} D_i : C_j(u) = 1 \right\}. \tag{5.15}$$

The table A_j contains all local assignments on the scope S_j that satisfy the constraint C_j.
Since

$$|S_j| \leq r, \tag{5.16}$$

one has

$$|A_j| \leq q^r. \tag{5.17}$$

The DRCC reduction object is the family of local admissible tables

$$\zeta = (A_1, A_2, \ldots, A_m). \tag{5.18}$$

This gives the finite collapse

$$q^n \longrightarrow (A_1, A_2, \ldots, A_m). \tag{5.19}$$

The cost of constructing the local tables is bounded by

$$N_{\mathrm{Collapse}}(P) \leq mq^r. \tag{5.20}$$

This is the first DRCC reduction step.

It is local and depends on the maximum constraint arity r, not on the full number of complete assignments q^n.

5.4 Reconstruction Space

Let

$$\pi_{S_j}(a) \tag{5.21}$$

denote the restriction of the complete assignment a to the scope S_j.

The reconstruction space is

$$\Omega_{\mathrm{CSP}}(\zeta) = \left\{a \in X_{\mathrm{CSP}}(P) : \pi_{S_j}(a) \in A_j \text{ for all } j = 1, \dots, m\right\}. \tag{5.22}$$

This is exactly the set of assignments that satisfy all constraints.

The reconstruction dimension is

$$d_{\mathrm{rec}}(P, \zeta) = |\Omega_{\mathrm{CSP}}(\zeta)|. \tag{5.23}$$

The DRCC structural stability condition becomes

$$0 < |\Omega_{\mathrm{CSP}}(\zeta)| \leq q^n < \infty. \tag{5.24}$$

If

$$|\Omega_{\mathrm{CSP}}(\zeta)| = 0, \tag{5.25}$$

then the CSP instance has no admissible reconstruction under the constraint system.

In that case, DRCC correctly reports an empty reconstruction space rather than a valid solution.

5.5 Controlled Reconstruction by Structural Width

To make the reconstruction phase explicit, assume that the constraint hypergraph admits a reconstruction ordering of width

$$\omega(P). \tag{5.26}$$

The width $\omega(P)$ measures the largest number of simultaneously active variables required during controlled reconstruction.

This quantity is not assumed to be small for all CSP instances.

It is an instance-dependent structural parameter.

Under a bounded-width reconstruction ordering, the number of partial states maintained during reconstruction is bounded by

$$q^{\omega(P)+1}. \tag{5.27}$$

Let

$$b(P) \tag{5.28}$$

denote the number of reconstruction steps.
Then a count-level reconstruction bound is

$$N_{\text{Reconstruction}}(P, \zeta) \leq b(P) q^{\omega(P)+1}. \tag{5.29}$$

This bound is conditional.
It applies only when such a bounded-width reconstruction ordering is available.

Proposition 5.1 (Controlled-Width DRCC Runtime Bound)**.** *Assume that a CSP instance P has maximum constraint arity r and admits a reconstruction ordering of width $\omega(P)$. Then the DRCC evaluation count satisfies*

$$N_{\text{DRCC}}(P, \zeta) \leq mq^r + b(P) q^{\omega(P)+1}. \tag{5.30}$$

Proof. By Equation (5.20), the cost of constructing all local constraint tables is at most

$$mq^r. \tag{5.31}$$

By Equation (5.29), the reconstruction cost under a bounded-width ordering is at most

$$b(P) q^{\omega(P)+1}. \tag{5.32}$$

The DRCC evaluation count is the sum of collapse and reconstruction costs.
Therefore,

$$N_{\text{DRCC}}(P, \zeta) \leq mq^r + b(P) q^{\omega(P)+1}. \tag{5.33}$$

This proves the claim. □

5.6 Runtime Model

The classical evaluation count is

$$N_{\text{Classic}}(P) = mq^n. \tag{5.34}$$

The DRCC evaluation count is bounded by

$$N_{\text{DRCC}}(P, \zeta) \leq mq^r + b(P) q^{\omega(P)+1}. \tag{5.35}$$

The corresponding runtimes are

$$T_{\text{Classic}}(P) = \frac{mq^n}{f} \tag{5.36}$$

and

$$T_{\text{DRCC}}(P, \zeta) = \frac{N_{\text{DRCC}}(P, \zeta)}{f}. \tag{5.37}$$

5.7 Runtime Gain

The runtime gain satisfies

$$G_{\text{CSP}}(P,\zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P,\zeta)}. \tag{5.38}$$

Using the runtime bound, one obtains the lower estimate

$$G_{\text{CSP}}(P,\zeta) \geq \frac{mq^n}{mq^r + b(P)q^{\omega(P)+1}}. \tag{5.39}$$

This expression shows the central structural condition.
DRCC becomes advantageous when

$$\omega(P) + 1 \ll n \tag{5.40}$$

and

$$r \ll n. \tag{5.41}$$

If the structural width grows with n in an uncontrolled way, then the DRCC gain may disappear.

5.8 Numerical Count-Level Example

Consider an illustrative CSP instance family with

$$q = 3, \qquad n = 20, \qquad m = 30, \qquad r = 3, \qquad \omega(P) = 4, \qquad b(P) = 30. \tag{5.42}$$

The classical count is

$$N_{\text{Classic}} = mq^n = 30 \cdot 3^{20}. \tag{5.43}$$

Thus,

$$N_{\text{Classic}} = 104{,}603{,}532{,}030. \tag{5.44}$$

The collapse count is

$$N_{\text{Collapse}} = mq^r = 30 \cdot 3^3 = 810. \tag{5.45}$$

The reconstruction count is bounded by

$$N_{\text{Reconstruction}} \leq b(P)q^{\omega(P)+1} = 30 \cdot 3^5 = 7{,}290. \tag{5.46}$$

Therefore,

$$N_{\text{DRCC}} \leq 810 + 7{,}290 = 8{,}100. \tag{5.47}$$

The resulting count-level gain is

$$G_{\text{CSP}} \geq \frac{104{,}603{,}532{,}030}{8{,}100} \approx 1.29 \cdot 10^7. \tag{5.48}$$

This example is conditional on the stated width model.
It is not a universal CSP speedup claim.

5.9 Evidence Table

Quantity	**Value**	**Interpretation**
q	3	Uniform domain size.
n	20	Number of variables.
m	30	Number of constraints.
r	3	Maximum constraint arity.
$\omega(P)$	4	Assumed reconstruction width.
N_{Classic}	104,603,532,030	Classical full-assignment count.
N_{DRCC}	$\leq 8{,}100$	DRCC count under the stated width assumption.
G	$\geq 1.29 \cdot 10^7$	Conditional count-level gain.

5.10 Transition Behavior

Assume for simplicity that

$$m = b(P) \tag{5.49}$$

and that q, r, and $\omega(P)$ remain fixed.
The transition condition is

$$mq^{n_w} = mq^r + mq^{\omega(P)+1}. \tag{5.50}$$

If $m > 0$, this reduces to

$$q^{n_w} = q^r + q^{\omega(P)+1}. \tag{5.51}$$

Thus,

$$n_w = \log_q \left(q^r + q^{\omega(P)+1}\right). \tag{5.52}$$

In the numerical example,

$$q = 3, \qquad r = 3, \qquad \omega(P) = 4. \tag{5.53}$$

Therefore,

$$n_w = \log_3 \left(3^3 + 3^5\right) = \log_3(270) \approx 5.10. \tag{5.54}$$

The nearest discrete transition occurs between

$$n = 5 \qquad \text{and} \qquad n = 6. \tag{5.55}$$

5.11 Finite Runtime Growth Interpretation

The classical CSP count is

$$N_{\text{Classic}}(P) = mq^n. \tag{5.56}$$

The DRCC count satisfies

$$N_{\text{DRCC}}(P, \zeta) \leq mq^r + b(P)q^{\omega(P)+1}. \tag{5.57}$$

The finite comparison is therefore

$$mq^n \quad \text{versus} \quad mq^r + b(P)q^{\omega(P)+1}. \tag{5.58}$$

The structural advantage appears when r and $\omega(P)$ remain small relative to n, and when the reconstruction ordering is constructively available.

This is a finite count-level statement.

It does not imply that arbitrary CSP instances become easy.

5.12 Structural Conditions

The CSP case is favorable for DRCC only under structural conditions.

The following conditions are required for a meaningful runtime gain:

1. Constraint arity r must remain small relative to n.
2. The reconstruction width $\omega(P)$ must remain controlled.
3. The reconstruction ordering must be constructively available.
4. Collapse overhead must be included in the DRCC count.
5. The reconstruction space must remain admissible and non-empty.

If these conditions fail, the DRCC model may lose its advantage.

5.13 Interpretation

The CSP case study shows that DRCC is closely related to structural decomposition.

The classical method considers all complete assignments.

The DRCC method constructs local admissible tables and reconstructs global solutions through controlled compatibility.

The key chain is

$$q^n \longrightarrow (A_1, \ldots, A_m) \longrightarrow \Omega_{\text{CSP}}(\zeta). \tag{5.59}$$

The runtime advantage is not automatic.

It depends on whether the structure of the constraints allows the reconstruction process to remain low-width.

5.14 Summary

For a uniform CSP with n variables, domain size q, m constraints, maximum arity r, and reconstruction width $\omega(P)$, the classical count is

$$N_{\text{Classic}} = mq^n. \tag{5.60}$$

The DRCC count satisfies

$$N_{\text{DRCC}} \leq mq^r + b(P)q^{\omega(P)+1}. \tag{5.61}$$

The gain estimate is

$$G_{\mathrm{CSP}} \geq \frac{mq^n}{mq^r + b(P)q^{\omega(P)+1}}. \tag{5.62}$$

The transition point under fixed structural parameters is

$$n_w = \log_q \left(q^r + q^{\omega(P)+1} \right). \tag{5.63}$$

The CSP example therefore supports a conditional DRCC claim:

$$\boxed{\text{Controlled reconstruction can reduce runtime when the constraint structure admits a bounded-width reconstruction process.}} \tag{5.64}$$

It does not support a universal claim that all CSP instances become easy under DRCC.

6 Boolean Satisfiability

6.1 Problem Definition

Let

$$\Phi \tag{6.1}$$

be a Boolean formula in conjunctive normal form.
Thus,

$$\Phi = K_1 \wedge K_2 \wedge \cdots \wedge K_m, \tag{6.2}$$

where each K_j is a clause.
Let

$$V = \{x_1, x_2, \ldots, x_n\} \tag{6.3}$$

be the set of Boolean variables.
Each variable has domain

$$D_i = \{0, 1\}. \tag{6.4}$$

A truth assignment is a map

$$a : V \to \{0, 1\}. \tag{6.5}$$

The Boolean satisfiability problem asks whether there exists an assignment a such that

$$\Phi(a) = 1. \tag{6.6}$$

The SAT case is included because it is a canonical exponential search problem.
The purpose here is not to solve SAT in the general worst-case sense.
The purpose is to express SAT instances in the DRCC language of controlled reduction, reconstruction spaces, runtime gain, and finite transition behavior.

6.2 Classical Boolean Enumeration

The classical candidate space is the set of all Boolean assignments:

$$X_{\mathrm{SAT}}(\Phi) = \{0,1\}^n. \tag{6.7}$$

Therefore,

$$d_{\mathrm{frag}}(\Phi) = |X_{\mathrm{SAT}}(\Phi)| = 2^n. \tag{6.8}$$

If every complete assignment is checked against all m clauses, then the classical evaluation count is

$$N_{\mathrm{Classic}}(\Phi) = m2^n. \tag{6.9}$$

The corresponding runtime is

$$T_{\mathrm{Classic}}(\Phi) = \frac{m2^n}{f}. \tag{6.10}$$

This is the exponential baseline.

6.3 Clause-Induced Reduction

DRCC does not begin by enumerating all Boolean assignments.

Instead, it constructs local admissible tables induced by the clauses.

For a clause K_j, let

$$S_j \subseteq V \tag{6.11}$$

be the set of variables appearing in K_j.

The clause arity is

$$|S_j|. \tag{6.12}$$

Let

$$r = \max_{1 \le j \le m} |S_j| \tag{6.13}$$

be the maximum clause width.

For each clause K_j, define the local admissible table

$$A_j = \{u \in \{0,1\}^{S_j} : K_j(u) = 1\}. \tag{6.14}$$

The reduced DRCC state is

$$\zeta = (A_1, A_2, \dots, A_m). \tag{6.15}$$

The construction of all local tables costs at most

$$N_{\mathrm{Collapse}}(\Phi) \le m2^r. \tag{6.16}$$

For a 3-SAT instance,

$$r = 3. \tag{6.17}$$

Hence each local clause table has at most

$$2^3 = 8 \tag{6.18}$$

local assignments before eliminating the falsifying local pattern.

6.4 Reconstruction Space of Admissible Assignments

Let

$$\pi_{S_j}(a) \tag{6.19}$$

denote the restriction of an assignment a to the variables in S_j.
The reconstruction space associated with the reduced state ζ is

$$\Omega_{\mathrm{SAT}}(\zeta) = \left\{a \in \{0,1\}^n : \pi_{S_j}(a) \in A_j \text{ for all } j = 1, \ldots, m\right\}. \tag{6.20}$$

Thus,

$$\Omega_{\mathrm{SAT}}(\zeta) = \{a \in \{0,1\}^n : \Phi(a) = 1\}. \tag{6.21}$$

The reconstruction dimension is

$$d_{\mathrm{rec}}(\Phi, \zeta) = |\Omega_{\mathrm{SAT}}(\zeta)|. \tag{6.22}$$

If

$$d_{\mathrm{rec}}(\Phi, \zeta) > 0, \tag{6.23}$$

then the instance is satisfiable.
If

$$d_{\mathrm{rec}}(\Phi, \zeta) = 0, \tag{6.24}$$

then the reconstruction space is empty, and no satisfying assignment is compatible with the clause tables.
For the task of reconstructing a satisfying assignment, the admissible solution condition is

$$0 < d_{\mathrm{rec}}(\Phi, \zeta) \leq 2^n < \infty. \tag{6.25}$$

For the decision version of SAT, the case $d_{\mathrm{rec}}(\Phi, \zeta) = 0$ is not a valid reconstruction, but it is a valid negative decision outcome if emptiness has been established by the reconstruction procedure.

6.5 Structural SAT Instances

DRCC does not assume that every SAT formula has a small reconstruction space.

A SAT instance is called structurally controlled if the reconstruction process can be performed with a bounded structural width.

Let

$$\omega(\Phi) \tag{6.26}$$

denote a reconstruction width parameter.

This parameter measures the maximum number of active variables that must be jointly maintained during controlled reconstruction.

The value of $\omega(\Phi)$ is instance-dependent.

It may be small for structured formulas and large for unstructured formulas.

If

$$\omega(\Phi) \approx n, \tag{6.27}$$

then the DRCC advantage may disappear.

6.6 Controlled Reconstruction

Assume that a reconstruction ordering is available.

Let

$$b(\Phi) \tag{6.28}$$

denote the number of reconstruction steps.

Under a reconstruction width bound $\omega(\Phi)$, the number of active partial assignments maintained at one step is bounded by

$$2^{\omega(\Phi)+1}. \tag{6.29}$$

Thus the reconstruction count is bounded by

$$N_{\text{Reconstruction}}(\Phi, \zeta) \leq b(\Phi) 2^{\omega(\Phi)+1}. \tag{6.30}$$

The DRCC count is therefore

$$N_{\text{DRCC}}(\Phi, \zeta) \leq m2^r + b(\Phi) 2^{\omega(\Phi)+1}. \tag{6.31}$$

Proposition 6.1 (Controlled Structural SAT Bound)**.** *Let Φ be a CNF formula with m clauses, maximum clause width r, and reconstruction width $\omega(\Phi)$.*

If a reconstruction ordering with $b(\Phi)$ steps is available, then

$$N_{\text{DRCC}}(\Phi, \zeta) \leq m2^r + b(\Phi) 2^{\omega(\Phi)+1}. \tag{6.32}$$

Proof. The collapse phase constructs all local clause tables.

By Equation (6.16), this costs at most

$$m2^r. \tag{6.33}$$

The reconstruction phase maintains at most

$$2^{\omega(\Phi)+1} \tag{6.34}$$

active partial assignments per reconstruction step.

With $b(\Phi)$ steps, this gives

$$b(\Phi)2^{\omega(\Phi)+1}. \tag{6.35}$$

Adding collapse and reconstruction costs gives

$$N_{\mathrm{DRCC}}(\Phi, \zeta) \leq m2^r + b(\Phi)2^{\omega(\Phi)+1}. \tag{6.36}$$

This proves the claim. □

6.7 Runtime Model

The classical count is

$$N_{\mathrm{Classic}}(\Phi) = m2^n. \tag{6.37}$$

The DRCC count satisfies

$$N_{\mathrm{DRCC}}(\Phi, \zeta) \leq m2^r + b(\Phi)2^{\omega(\Phi)+1}. \tag{6.38}$$

The corresponding runtimes are

$$T_{\mathrm{Classic}}(\Phi) = \frac{m2^n}{f} \tag{6.39}$$

and

$$T_{\mathrm{DRCC}}(\Phi, \zeta) = \frac{N_{\mathrm{DRCC}}(\Phi, \zeta)}{f}. \tag{6.40}$$

6.8 Runtime Gain

The runtime gain is

$$G_{\mathrm{SAT}}(\Phi, \zeta) = \frac{N_{\mathrm{Classic}}(\Phi)}{N_{\mathrm{DRCC}}(\Phi, \zeta)}. \tag{6.41}$$

Using the DRCC bound, one obtains

$$G_{\mathrm{SAT}}(\Phi, \zeta) \geq \frac{m2^n}{m2^r + b(\Phi)2^{\omega(\Phi)+1}}. \tag{6.42}$$

This expression shows the conditional nature of the runtime gain.
DRCC becomes advantageous only if

$$r \ll n \tag{6.43}$$

and

$$\omega(\Phi) \ll n. \tag{6.44}$$

6.9 Numerical Count-Level Example

Consider an illustrative structured 3-SAT family with

$$n = 40, \qquad m = 120, \qquad r = 3, \qquad \omega(\Phi) = 6, \qquad b(\Phi) = 120. \tag{6.45}$$

The classical count is

$$N_{\text{Classic}} = 120 \cdot 2^{40}. \tag{6.46}$$

Thus,

$$N_{\text{Classic}} = 131{,}941{,}395{,}333{,}120. \tag{6.47}$$

The collapse count is

$$N_{\text{Collapse}} = 120 \cdot 2^{3} = 960. \tag{6.48}$$

The reconstruction count is bounded by

$$N_{\text{Reconstruction}} \leq 120 \cdot 2^{7} = 15{,}360. \tag{6.49}$$

Thus,

$$N_{\text{DRCC}} \leq 960 + 15{,}360 = 16{,}320. \tag{6.50}$$

The count-level runtime gain is

$$G_{\text{SAT}} \geq \frac{131{,}941{,}395{,}333{,}120}{16{,}320} \approx 8.08 \cdot 10^{9}. \tag{6.51}$$

This result is conditional on the stated reconstruction width.
It is not a worst-case SAT result.

6.10 A Small 3-SAT Structural Example

To illustrate structural collapse explicitly, consider a 3-SAT formula with variables

$$V = \{x_1, x_2, x_3\}. \tag{6.52}$$

Let

$$\Phi_3 = K_1 \wedge K_2 \wedge K_3 \tag{6.53}$$

with

$$K_1 = (x_1 \vee x_2 \vee \neg x_3), \tag{6.54}$$
$$K_2 = (\neg x_1 \vee x_2 \vee x_3), \tag{6.55}$$
$$K_3 = (x_1 \vee \neg x_2 \vee x_3). \tag{6.56}$$

The classical assignment space has size

$$|X_{\text{SAT}}(\Phi_3)| = 2^{3} = 8. \tag{6.57}$$

For each assignment a, define the clause-output vector

$$\sigma_{\Phi_3}(a) = (K_1(a), K_2(a), K_3(a)). \tag{6.58}$$

Two assignments are structurally equivalent if they produce the same clause-output vector:

$$a \sim_{\Phi_3} b \iff \sigma_{\Phi_3}(a) = \sigma_{\Phi_3}(b). \tag{6.59}$$

The DRCC rank of the clause-output structure is

$$R_{\mathrm{DRCC}}(\Phi_3) = \left|\{\sigma_{\Phi_3}(a) : a \in \{0,1\}^3\}\right|. \tag{6.60}$$

For this example, the eight assignments collapse into four distinct clause-output classes:

$$8 \longrightarrow 4. \tag{6.61}$$

This shows structural compression of clause-output behavior.
It does not imply that arbitrary SAT instances admit the same collapse.

6.11 Evidence Table

Quantity	**Value**	**Interpretation**
n	40	Number of Boolean variables.
m	120	Number of clauses.
r	3	Maximum clause width.
$\omega(\Phi)$	6	Assumed reconstruction width.
N_{Classic}	131,941,395 333,120	Classical full-assignment count.
N_{DRCC}	$\leq$ 16,320	DRCC count under structural-width assumption.
G	$\geq 8.08 \cdot 10^9$	Conditional count-level runtime gain.
Small 3-SAT	$8 \to 4$	Assignments collapse into four clause-output classes.

6.12 Transition Point

Assume for simplicity that

$$b(\Phi) = m \tag{6.62}$$

and that r and $\omega(\Phi)$ are fixed.
The transition condition is

$$m2^{n_w} = m2^r + m2^{\omega(\Phi)+1}. \tag{6.63}$$

If $m > 0$, this becomes

$$2^{n_w} = 2^r + 2^{\omega(\Phi)+1}. \tag{6.64}$$

Therefore,

$$n_w = \log_2\left(2^r + 2^{\omega(\Phi)+1}\right). \tag{6.65}$$

For the example values

$$r = 3, \qquad \omega(\Phi) = 6, \tag{6.66}$$

one obtains

$$n_w = \log_2(2^3 + 2^7) = \log_2(136) \approx 7.09. \tag{6.67}$$

Thus the nearest discrete transition occurs between

$$n = 7 \qquad \text{and} \qquad n = 8. \tag{6.68}$$

6.13 Finite SAT Runtime Interpretation

The classical SAT count is

$$N_{\text{Classic}}(\Phi) = m2^n. \tag{6.69}$$

The DRCC count satisfies

$$N_{\text{DRCC}}(\Phi, \zeta) \leq m2^r + b(\Phi)2^{\omega(\Phi)+1}. \tag{6.70}$$

Thus, the finite comparison is

$$m2^n \quad \text{versus} \quad m2^r + b(\Phi)2^{\omega(\Phi)+1}. \tag{6.71}$$

If r, $\omega(\Phi)$, m, and $b(\Phi)$ remain controlled relative to n, then the DRCC count may be significantly smaller than full Boolean enumeration.

This is a structural runtime statement.

It is not a complexity-theoretic collapse statement.

6.14 Non-Claim: No General Proof of P versus NP

This SAT chapter does not prove

$$P = NP \tag{6.72}$$

and does not prove

$$P \neq NP. \tag{6.73}$$

It does not claim a polynomial-time algorithm for arbitrary SAT instances.

It does not replace DPLL, CDCL, resolution, local search, or established SAT-solving methods.

The claim is narrower:

> For structurally controlled SAT instances with bounded reconstruction width, DRCC provides a count-level runtime model in which controlled local reduction and reconstruction may be significantly smaller than full Boolean enumeration. (6.74)

If the reconstruction width grows with n, then the DRCC runtime advantage may disappear.

6.15 Summary

For a CNF formula with n variables and m clauses, the classical candidate count is

$$N_{\text{Classic}} = m2^n. \tag{6.75}$$

The DRCC count satisfies

$$N_{\text{DRCC}} \leq m2^r + b(\Phi)2^{\omega(\Phi)+1}. \tag{6.76}$$

The runtime gain satisfies

$$G_{\text{SAT}} \geq \frac{m2^n}{m2^r + b(\Phi)2^{\omega(\Phi)+1}}. \tag{6.77}$$

The transition point under fixed structural parameters is

$$n_w = \log_2 \left(2^r + 2^{\omega(\Phi)+1}\right). \tag{6.78}$$

The small 3-SAT example illustrates the finite structural compression

$$8 \longrightarrow 4. \tag{6.79}$$

The SAT case therefore supports a conditional DRCC runtime statement for structurally controlled formulas, while explicitly avoiding any claim of a general SAT solution or a resolution of the P versus NP problem.

7 Graph 3-Coloring

7.1 Problem Definition

Let

$$G = (V, E) \tag{7.1}$$

be a finite undirected graph with

$$|V| = n \qquad \text{and} \qquad |E| = m. \tag{7.2}$$

Let

$$K = \{1, 2, 3\} \tag{7.3}$$

be the set of available colors.
A coloring is a map

$$c : V \to K. \tag{7.4}$$

A coloring c is proper if

$$c(u) \neq c(v) \qquad \text{for every edge } \{u, v\} \in E. \tag{7.5}$$

The graph 3-coloring problem asks whether there exists at least one proper coloring of G.

The problem is included as a case study because it has a simple classical search space and a natural local constraint structure.

7.2 Classical Coloring Space

The classical candidate space is

$$X_{\mathrm{GC}}(G) = K^V. \tag{7.6}$$

Thus,

$$d_{\mathrm{frag}}(G) = |X_{\mathrm{GC}}(G)| = 3^n. \tag{7.7}$$

If every complete coloring is checked against every edge, the classical evaluation count is

$$N_{\mathrm{Classic}}(G) = m3^n. \tag{7.8}$$

The corresponding runtime is

$$T_{\mathrm{Classic}}(G) = \frac{m3^n}{f}. \tag{7.9}$$

This is the exponential baseline for graph coloring.

7.3 DRCC Collapse by Edge Constraints

DRCC does not begin by enumerating all colorings.
Instead, it constructs local admissible edge tables.
For each edge

$$e = \{u, v\} \in E, \tag{7.10}$$

define

$$A_e = \{(a, b) \in K^2 : a \neq b\}. \tag{7.11}$$

Since K has three colors,

$$|A_e| = 6. \tag{7.12}$$

The reduced DRCC state is the family of all local edge tables:

$$\zeta = (A_e)_{e \in E}. \tag{7.13}$$

The collapse cost is bounded by

$$N_{\mathrm{Collapse}}(G) \leq 6m. \tag{7.14}$$

This is the first finite reduction step:

$$3^n \longrightarrow (A_e)_{e \in E}. \tag{7.15}$$

The reduction records local admissibility structure without enumerating all complete colorings.

7.4 Reconstruction Space

The reconstruction space associated with ζ is

$$\Omega_{\mathrm{GC}}(\zeta) = \left\{ c \in K^V : (c(u), c(v)) \in A_e \text{ for every } e = \{u, v\} \in E \right\}. \tag{7.16}$$

Thus,

$$\Omega_{\mathrm{GC}}(\zeta) = \{c \in K^V : c \text{ is a proper 3-coloring of } G\}. \tag{7.17}$$

The reconstruction dimension is

$$d_{\mathrm{rec}}(G, \zeta) = |\Omega_{\mathrm{GC}}(\zeta)|. \tag{7.18}$$

If

$$d_{\mathrm{rec}}(G, \zeta) > 0, \tag{7.19}$$

then G is 3-colorable.
If

$$d_{\mathrm{rec}}(G, \zeta) = 0, \tag{7.20}$$

then no admissible reconstruction exists.
For the reconstruction task, the structural stability condition is

$$0 < d_{\mathrm{rec}}(G, \zeta) \leq 3^n < \infty. \tag{7.21}$$

For the decision task, an empty reconstruction space is a valid negative decision outcome if emptiness is established by the reconstruction procedure.

7.5 Finite Reconstruction Fiber Interpretation

The reconstruction space may be viewed as a finite reconstruction fiber:

$$\mathrm{RF}(G, \zeta) = \Omega_{\mathrm{GC}}(\zeta). \tag{7.22}$$

This notation means only the finite set of all colorings consistent with the reduced edge-state ζ.
It does not imply any differentiable or continuous geometric structure.
The point is finite and combinatorial:

$$K^V \longrightarrow (A_e)_{e \in E} \longrightarrow \Omega_{\mathrm{GC}}(\zeta). \tag{7.23}$$

7.6 Controlled Reconstruction

A controlled reconstruction procedure builds colorings from the local edge tables while maintaining only a limited number of active vertices.
Let

$$\omega(G) \tag{7.24}$$

denote a reconstruction width parameter.

This parameter measures the maximum number of simultaneously active vertices required during reconstruction.

Let

$$b(G) \tag{7.25}$$

denote the number of reconstruction steps.

Under a bounded-width reconstruction ordering, the number of active partial colorings is bounded by

$$3^{\omega(G)+1}. \tag{7.26}$$

Therefore,

$$N_{\text{Reconstruction}}(G, \zeta) \leq b(G)3^{\omega(G)+1}. \tag{7.27}$$

The DRCC count satisfies

$$N_{\text{DRCC}}(G, \zeta) \leq 6m + b(G)3^{\omega(G)+1}. \tag{7.28}$$

Proposition 7.1 (Controlled Structural Coloring Bound)**.** *Let $G = (V, E)$ be a finite graph with $m = |E|$.*

If a reconstruction ordering with width $\omega(G)$ and $b(G)$ steps is available, then

$$N_{\text{DRCC}}(G, \zeta) \leq 6m + b(G)3^{\omega(G)+1}. \tag{7.29}$$

Proof. For each edge $e \in E$, the admissible local table A_e contains six ordered color pairs.

Hence, the cost of constructing all local edge tables is bounded by

$$6m. \tag{7.30}$$

During reconstruction, the bounded-width assumption allows at most

$$3^{\omega(G)+1} \tag{7.31}$$

active partial colorings per reconstruction step.

With $b(G)$ reconstruction steps, the reconstruction count is bounded by

$$b(G)3^{\omega(G)+1}. \tag{7.32}$$

Adding the collapse and reconstruction terms gives

$$N_{\text{DRCC}}(G, \zeta) \leq 6m + b(G)3^{\omega(G)+1}. \tag{7.33}$$

This proves the claim. □

7.7 Reconstruction Paths

If a graph structure is placed on the set of partial reconstructions, a reconstruction path is a finite sequence

$$\gamma = (c_0, c_1, \ldots, c_\ell), \tag{7.34}$$

where each c_i is a partial coloring and each step adds or modifies one admissible local decision. The length of the path is

$$\text{length}(\gamma) = \ell. \tag{7.35}$$

A shortest admissible reconstruction path ending at a proper coloring $c \in \Omega_{\text{GC}}(\zeta)$ may be written as

$$\gamma^*(G, \zeta, c) = \arg\min_{\gamma} \text{length}(\gamma). \tag{7.36}$$

This path viewpoint measures reconstruction cost inside the reconstruction space.

It is not required for every runtime estimate, but it provides a finite structural interpretation of controlled reconstruction.

7.8 Runtime Gain

The runtime gain is

$$G_{\text{GC}}(G, \zeta) = \frac{N_{\text{Classic}}(G)}{N_{\text{DRCC}}(G, \zeta)}. \tag{7.37}$$

Using the runtime bound gives

$$G_{\text{GC}}(G, \zeta) \geq \frac{m3^n}{6m + b(G)3^{\omega(G)+1}}. \tag{7.38}$$

The gain is significant only when

$$\omega(G) \ll n. \tag{7.39}$$

If $\omega(G)$ grows with n in an uncontrolled way, then the DRCC advantage may disappear.

7.9 Numerical Count-Level Example

Consider an illustrative graph with

$$n = 24, \qquad m = 36, \qquad \omega(G) = 4, \qquad b(G) = 24. \tag{7.40}$$

The classical count is

$$N_{\text{Classic}} = 36 \cdot 3^{24}. \tag{7.41}$$

Thus,

$$N_{\text{Classic}} = 10{,}167{,}463{,}313{,}316. \tag{7.42}$$

The collapse count is

$$N_{\text{Collapse}} = 6m = 216. \tag{7.43}$$

The reconstruction count is bounded by

$$N_{\text{Reconstruction}} \leq 24 \cdot 3^5 = 5{,}832. \tag{7.44}$$

Thus,

$$N_{\text{DRCC}} \leq 216 + 5{,}832 = 6{,}048. \tag{7.45}$$

The count-level gain is

$$G_{\text{GC}} \geq \frac{10{,}167{,}463{,}313{,}316}{6{,}048} \approx 1.68 \cdot 10^9. \tag{7.46}$$

This number is conditional on the assumed reconstruction width.
It is not a universal graph-coloring speedup claim.

7.10 Small Four-Vertex Cycle Example

To show the reconstruction space explicitly, consider the cycle graph

$$C_4 = (V, E) \tag{7.47}$$

with

$$V = \{v_1, v_2, v_3, v_4\} \tag{7.48}$$

and

$$E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}\}. \tag{7.49}$$

The classical coloring space has size

$$|K^V| = 3^4 = 81. \tag{7.50}$$

Suppose a partial reduced state fixes

$$c(v_1) = 1, \qquad c(v_2) = 2. \tag{7.51}$$

The remaining vertices v_3 and v_4 must satisfy the edge constraints

$$c(v_3) \neq c(v_2), \qquad c(v_4) \neq c(v_3), \qquad c(v_4) \neq c(v_1). \tag{7.52}$$

A direct finite check gives the admissible completions

$$(c(v_3), c(v_4)) \in \{(1,2), (1,3), (3,2)\}. \tag{7.53}$$

Hence,

$$d_{\text{rec}}(C_4, \zeta) = 3. \tag{7.54}$$

The local fragment therefore reduces the full coloring space from

$$81 \tag{7.55}$$

to the reconstruction count

$$3. \tag{7.56}$$

This example is finite, explicit, and reproducible.
It does not imply a general polynomial-time algorithm for arbitrary graph coloring.

7.11 Evidence Table

Quantity	**Value**	**Interpretation**
n	24	Number of vertices in the numerical example.
m	36	Number of edges in the numerical example.
$\omega(G)$	4	Assumed reconstruction width.
d_{frag}	3^{24}	Classical coloring space.
N_{Classic}	10,167,463 313,316	Classical edge-check count.
N_{DRCC}	$\leq$ 6,048	DRCC count under width assumption.
G	$\geq 1.68 \cdot 10^9$	Conditional count-level gain.
C_4 example	$81 \to 3$	Explicit finite reconstruction after fixing a local fragment.

7.12 Transition Point

Assume for simplicity that $b(G)$ and m scale proportionally and that $\omega(G)$ remains fixed.
The transition condition is

$$m3^{n_w} = 6m + b(G)3^{\omega(G)+1}. \tag{7.57}$$

If $m > 0$, then

$$3^{n_w} = 6 + \frac{b(G)}{m}3^{\omega(G)+1}. \tag{7.58}$$

Thus,

$$n_w = \log_3\left(6 + \frac{b(G)}{m}3^{\omega(G)+1}\right). \tag{7.59}$$

For the numerical example,

$$\frac{b(G)}{m} = \frac{24}{36} = \frac{2}{3}, \qquad \omega(G) = 4. \tag{7.60}$$

Hence,

$$n_w = \log_3\left(6 + \frac{2}{3}3^5\right) = \log_3(168) \approx 4.66. \tag{7.61}$$

The nearest discrete transition is therefore around

$$n = 5. \tag{7.62}$$

7.13 Interpretation

The graph-coloring case illustrates a finite reconstruction-space viewpoint.
Classical search enumerates all colorings in

$$K^V. \tag{7.63}$$

DRCC constructs local admissibility tables for edges and reconstructs proper colorings from compatible local decisions.
The core chain is

$$3^n \longrightarrow (A_e)_{e\in E} \longrightarrow \Omega_{\mathrm{GC}}(\zeta). \tag{7.64}$$

The runtime advantage depends on structural width.
Thus, the graph-coloring example supports a conditional claim:

$$\boxed{\text{Controlled reconstruction can reduce enumeration cost for graph-coloring instances whose constraint structure admits bounded-width reconstruction.}} \tag{7.65}$$

It does not imply that arbitrary graph coloring becomes easy.

7.14 Summary

For graph 3-coloring, the classical count is

$$N_{\mathrm{Classic}}(G) = m3^n. \tag{7.66}$$

The DRCC count satisfies

$$N_{\mathrm{DRCC}}(G,\zeta) \leq 6m + b(G)3^{\omega(G)+1}. \tag{7.67}$$

The gain estimate is

$$G_{\mathrm{GC}}(G,\zeta) \geq \frac{m3^n}{6m + b(G)3^{\omega(G)+1}}. \tag{7.68}$$

The small C_4 example gives the explicit finite reconstruction result

$$81 \longrightarrow 3. \tag{7.69}$$

The graph 3-coloring case therefore provides both a general width-conditional runtime model and a small fully explicit reconstruction example.

8 Full Adder

8.1 Problem Definition

The full adder is a finite Boolean reconstruction problem.
Let

$$B = \{0, 1\} \tag{8.1}$$

be the Boolean alphabet.
The input space of the full adder is

$$X_{\text{FA}} = B^3. \tag{8.2}$$

An element of X_{FA} is written as

$$x = (a, b, c_{\text{in}}), \tag{8.3}$$

where a and b are the two input bits and c_{in} is the carry-in bit.
Thus,

$$|X_{\text{FA}}| = 2^3 = 8. \tag{8.4}$$

The full adder computes two output bits:

$$s \qquad \text{and} \qquad c_{\text{out}}, \tag{8.5}$$

where s is the sum bit and c_{out} is the carry-out bit.
The full adder map is

$$F_{\text{FA}} : B^3 \to B^2. \tag{8.6}$$

It is defined by

$$F_{\text{FA}}(a, b, c_{\text{in}}) = (s, c_{\text{out}}). \tag{8.7}$$

The output bits are given by

$$s = a \oplus b \oplus c_{\text{in}}, \tag{8.8}$$

and

$$c_{\text{out}} = ab \vee ac_{\text{in}} \vee bc_{\text{in}}. \tag{8.9}$$

Equivalently, if

$$h(x) = a + b + c_{\text{in}}, \tag{8.10}$$

then

$$s = h(x) \bmod 2 \tag{8.11}$$

and

$$c_{\text{out}} = \begin{cases} 0, & h(x) \in \{0,1\}, \\ 1, & h(x) \in \{2,3\}. \end{cases} \tag{8.12}$$

The full adder is included as a micro-case because it is finite, complete, reproducible, and small enough to show the entire DRCC collapse explicitly.

8.2 Classical Candidate Space

The classical candidate space is the full Boolean cube

$$X^{\text{FA}}_{\text{Classic}} = B^3. \tag{8.13}$$

Therefore,

$$d_{\text{frag}}(X_{\text{FA}}) = |X^{\text{FA}}_{\text{Classic}}| = 8. \tag{8.14}$$

If each input vector is evaluated separately, the classical evaluation count is

$$N^{\text{FA}}_{\text{Classic}} = 8. \tag{8.15}$$

The corresponding runtime is

$$T^{\text{FA}}_{\text{Classic}} = \frac{8}{f}, \tag{8.16}$$

where f denotes the evaluation frequency or elementary operation rate.

This is not a large runtime.

The purpose of the full adder case is not to demonstrate a large speedup, but to show a complete controlled collapse in a transparent finite setting.

8.3 DRCC Collapse by Hamming Weight

DRCC does not treat all eight Boolean inputs as structurally distinct.

The relevant structural quantity is the Hamming weight

$$h : B^3 \to \{0,1,2,3\}, \tag{8.17}$$

defined by

$$h(a, b, c_{\text{in}}) = a + b + c_{\text{in}}. \tag{8.18}$$

The controlled reduction map is

$$C_{\text{FA}} : B^3 \to \{0,1,2,3\}. \tag{8.19}$$

It is defined by

$$C_{\text{FA}}(x) = h(x). \tag{8.20}$$

Thus, the eight Boolean input states collapse into four structural classes.

For each $k \in \{0,1,2,3\}$, define

$$C_k = \{x \in B^3 : h(x) = k\}. \tag{8.21}$$

The family of structural classes is

$$\mathcal{C}_{\text{FA}} = \{C_0, C_1, C_2, C_3\}. \tag{8.22}$$

The class sizes are

$$|C_0| = 1, \qquad |C_1| = 3, \qquad |C_2| = 3, \qquad |C_3| = 1. \tag{8.23}$$

The DRCC reduced state is

$$\zeta_{\text{FA}} = \mathcal{C}_{\text{FA}}. \tag{8.24}$$

The collapse chain is

$$8 \longrightarrow 4. \tag{8.25}$$

Equivalently,

$$B^3 \longrightarrow \{C_0, C_1, C_2, C_3\}. \tag{8.26}$$

The collapse count is bounded by the number of input states inspected:

$$N^{\text{FA}}_{\text{Collapse}} \leq 8. \tag{8.27}$$

If the Hamming weight is computed directly from the three bits, the elementary bit-addition count is bounded by

$$N^{\text{FA}}_{\text{Collapse,bit}} \leq 3 \cdot 8 = 24. \tag{8.28}$$

The first count measures state inspection.
The second count measures a more explicit bit-level implementation.

8.4 Output Reconstruction from Structural Classes

The output of the full adder depends only on the Hamming weight.
Define the class-output map

$$Q_{\text{FA}} : \{0, 1, 2, 3\} \to B^2. \tag{8.29}$$

It is defined by

$$Q_{\text{FA}}(k) = (k \bmod 2, \mathbf{1}_{\{k \geq 2\}}), \tag{8.30}$$

where $\mathbf{1}_{\{k \geq 2\}}$ denotes the indicator function of the condition $k \geq 2$.
Thus,

$$F_{\text{FA}} = Q_{\text{FA}} \circ C_{\text{FA}}. \tag{8.31}$$

This identity is the central DRCC factorization for the full adder case.

It says that the full adder output can be reconstructed from the collapsed Hamming-weight class.

The explicit class-output table is:

Class	**Hamming Weight**	**Sum Bit**	**Carry-Out Bit**
C_0	0	0	0
C_1	1	1	0
C_2	2	0	1
C_3	3	1	1

This table is the finite reconstruction table of the full adder.

8.5 Reconstruction Space

For a reduced class value

$$k \in \{0, 1, 2, 3\}, \tag{8.32}$$

the reconstruction fiber is

$$\Omega_{\mathrm{FA}}(k) = \{x \in B^3 : C_{\mathrm{FA}}(x) = k\}. \tag{8.33}$$

Since $C_{\mathrm{FA}}(x) = h(x)$, this can also be written as

$$\Omega_{\mathrm{FA}}(k) = C_k. \tag{8.34}$$

The reconstruction dimension of a class is

$$d_{\mathrm{rec}}^{\mathrm{FA}}(k) = |\Omega_{\mathrm{FA}}(k)|. \tag{8.35}$$

Therefore,

$$d_{\mathrm{rec}}^{\mathrm{FA}}(0) = 1, \qquad d_{\mathrm{rec}}^{\mathrm{FA}}(1) = 3, \qquad d_{\mathrm{rec}}^{\mathrm{FA}}(2) = 3, \qquad d_{\mathrm{rec}}^{\mathrm{FA}}(3) = 1. \tag{8.36}$$

The total reduced structural dimension is

$$R_{\mathrm{DRCC}}^{\mathrm{FA}} = |\mathcal{C}_{\mathrm{FA}}| = 4. \tag{8.37}$$

The stability condition for each non-empty reconstruction fiber is

$$0 < d_{\mathrm{rec}}^{\mathrm{FA}}(k) \le d_{\mathrm{frag}}(X_{\mathrm{FA}}) = 8 < \infty. \tag{8.38}$$

Since every class C_k is non-empty, the full adder satisfies the finite DRCC stability condition for all four structural classes.

8.6 Controlled Reconstruction Operator

The controlled reconstruction operator acts on a reduced class value and returns the corresponding output pair.

Define

$$R_{\mathrm{FA}} : \{0, 1, 2, 3\} \to B^2. \tag{8.39}$$

It is defined by

$$R_{\mathrm{FA}}(k) = Q_{\mathrm{FA}}(k). \tag{8.40}$$

Thus,

$$R_{\mathrm{FA}}(0) = (0,0), \tag{8.41}$$

$$R_{\mathrm{FA}}(1) = (1,0), \tag{8.42}$$

$$R_{\mathrm{FA}}(2) = (0,1), \tag{8.43}$$

and

$$R_{\mathrm{FA}}(3) = (1,1). \tag{8.44}$$

For every input $x \in B^3$, the original full adder output is recovered by

$$F_{\mathrm{FA}}(x) = R_{\mathrm{FA}}(C_{\mathrm{FA}}(x)). \tag{8.45}$$

This identity shows that the reduction is lossless with respect to the full adder output.
It does not preserve the identity of the original input state inside a class.
It preserves only the information required to reconstruct the output.

8.7 Algorithmic Protocol

The full adder case admits a direct DRCC-style protocol.

Step	**Operation**
1	Read the input state $x = (a, b, c_{\mathrm{in}}) \in B^3$.
2	Compute the structural class $k = C_{\mathrm{FA}}(x) = a + b + c_{\mathrm{in}}$.
3	Use the reconstruction operator $R_{\mathrm{FA}}(k)$.
4	Return the output pair $(s, c_{\mathrm{out}}) = R_{\mathrm{FA}}(k)$.

Equivalently, the protocol can be written as finite pseudocode.

Input: $a, b, c_{\mathrm{in}} \in \{0, 1\}$.
$k := a + b + c_{\mathrm{in}}$.
If $k = 0$, return $(0, 0)$.
If $k = 1$, return $(1, 0)$.
If $k = 2$, return $(0, 1)$.
If $k = 3$, return $(1, 1)$.

This protocol is fully reproducible because all input states, reduction classes, and output states are finite and explicitly defined.

8.8 Runtime Count

The DRCC runtime count separates collapse and reconstruction.
The collapse step computes

$$k = a + b + c_{\mathrm{in}}. \tag{8.46}$$

This requires a constant number of elementary operations.
Thus,

$$N^{\mathrm{FA}}_{\mathrm{Collapse}}(x) \leq 3. \tag{8.47}$$

The reconstruction step performs one table lookup or one class-output evaluation:

$$N^{\mathrm{FA}}_{\mathrm{Reconstruction}}(x) \leq 1. \tag{8.48}$$

Therefore, for one input state,

$$N^{\mathrm{FA}}_{\mathrm{DRCC}}(x) \leq 4. \tag{8.49}$$

For all eight input states, the complete count is bounded by

$$N^{\mathrm{FA}}_{\mathrm{DRCC}} \leq 8 \cdot 4 = 32. \tag{8.50}$$

This count is an implementation-level bound.
It is not directly smaller than the symbolic classical state count 8, because it counts different elementary operations.
For a fair structural comparison, DRCC counts the number of distinct output-relevant classes rather than the number of raw inputs.
The structural DRCC count is

$$N^{\mathrm{FA}}_{\mathrm{DRCC,struct}} = 4. \tag{8.51}$$

The corresponding structural gain is

$$G_{\mathrm{FA,struct}} = \frac{N^{\mathrm{FA}}_{\mathrm{Classic}}}{N^{\mathrm{FA}}_{\mathrm{DRCC,struct}}} = \frac{8}{4} = 2. \tag{8.52}$$

This is a structural reduction factor, not a claim of hardware-level speedup.

8.9 Evidence Table

Quantity	**Value**	**Interpretation**
Input space	B^3	All Boolean input triples.
$\lvert X_{\mathrm{FA}} \rvert$	8	Classical candidate states.
Reduction map	$C_{\mathrm{FA}}(x) = h(x)$	Collapse by Hamming weight.
Structural classes	C_0, C_1, C_2, C_3	Four output-relevant classes.
$R^{\mathrm{FA}}_{\mathrm{DRCC}}$	4	Number of distinct structural classes.
d_{frag}	8	Fragment space before collapse.
$d_{\mathrm{rec}}(k)$	$1, 3, 3, 1$	Fiber sizes after collapse.
$G_{\mathrm{FA,struct}}$	2	Structural reduction factor.

8.10 Transition Interpretation

For the full adder, the transition point is not asymptotic.
The input size is fixed at three Boolean variables.
Thus, the transition point is interpreted as a finite structural threshold:

$$|X_{\mathrm{FA}}| = 8 \qquad \text{and} \qquad |\mathcal{C}_{\mathrm{FA}}| = 4. \tag{8.53}$$

The finite transition condition is

$$|\mathcal{C}_{\mathrm{FA}}| < |X_{\mathrm{FA}}|. \tag{8.54}$$

Since

$$4 < 8, \tag{8.55}$$

the full adder admits a nontrivial output-preserving collapse.

This transition is not a runtime phase transition in the asymptotic sense.

It is a finite demonstration that output-relevant information may have smaller structural dimension than the raw input space.

8.11 DRCC-Gate Interpretation

The full adder can also be interpreted as a DRCC-gate.

A DRCC-gate is a finite logical operator whose output is reconstructed from a reduced structural class rather than from full raw enumeration.

For the full adder, the raw input space is

$$B^3. \tag{8.56}$$

The reduced gate-state space is

$$\{0, 1, 2, 3\}. \tag{8.57}$$

The gate reconstruction is

$$\{0, 1, 2, 3\} \longrightarrow B^2. \tag{8.58}$$

Thus, the DRCC-gate chain is

$$B^3 \longrightarrow \{0, 1, 2, 3\} \longrightarrow B^2. \tag{8.59}$$

This gives a finite example of output-preserving structural reconstruction.

It does not imply that arbitrary Boolean circuits admit the same collapse.

8.12 Interpretation

The full adder case illustrates the simplest complete DRCC pattern.

The classical view distinguishes all eight Boolean input states:

$$B^3. \tag{8.60}$$

The DRCC view collapses these states into four Hamming-weight classes:

$$C_0, C_1, C_2, C_3. \tag{8.61}$$

The output is reconstructed from the class value rather than from the identity of the individual input state.

The central chain is

$$B^3 \longrightarrow \{0,1,2,3\} \longrightarrow B^2. \tag{8.62}$$

Equivalently,

$$x \longrightarrow C_{\mathrm{FA}}(x) \longrightarrow R_{\mathrm{FA}}(C_{\mathrm{FA}}(x)). \tag{8.63}$$

This demonstrates the DRCC principle in a finite and reproducible form:

$$\boxed{\begin{array}{l}\text{Preserve the information required for reconstruction and collapse the input}\\ \text{distinctions that are irrelevant to the required output.}\end{array}} \tag{8.64}$$

For the full adder, the collapsed Hamming-weight class is sufficient to reconstruct the output pair.

It is not sufficient to reconstruct the original input uniquely.

Thus, the reduction is output-preserving but not input-invertible.

8.13 Limit of the Full Adder Case

The full adder is a micro-case.

It does not establish a general complexity-theoretic result.

It does not imply that arbitrary Boolean circuits admit a comparable collapse.

It only shows that, for this specific finite Boolean reconstruction problem, the output-relevant structure is smaller than the full input space.

This limitation is important because DRCC-V2 does not claim a universal algorithmic speedup.

The full adder provides a controlled example of structural compression, not a proof of broad computational tractability.

8.14 Summary

For the full adder, the classical candidate space is

$$X_{\mathrm{FA}} = B^3. \tag{8.65}$$

The fragment dimension is

$$d_{\mathrm{frag}}(X_{\mathrm{FA}}) = 8. \tag{8.66}$$

The controlled reduction map is

$$C_{\mathrm{FA}}(x) = h(x). \tag{8.67}$$

The reduced structural class space is

$$\mathcal{C}_{\mathrm{FA}} = \{C_0, C_1, C_2, C_3\}. \tag{8.68}$$

The number of DRCC structural classes is

$$R^{\mathrm{FA}}_{\mathrm{DRCC}} = 4. \tag{8.69}$$

The reconstruction identity is

$$F_{\mathrm{FA}}(x) = R_{\mathrm{FA}}(C_{\mathrm{FA}}(x)). \tag{8.70}$$

The structural gain is

$$G_{\mathrm{FA,struct}} = 2. \tag{8.71}$$

The full adder case therefore gives a complete finite example of controlled collapse, reconstruction by a reduced class, and explicit runtime accounting.

9 Cross-Case Runtime Analysis

9.1 One-Page Cross-Case Comparison

This chapter compares the finite runtime structure of the case studies developed in the previous chapters.

The purpose is not to prove a universal speedup theorem.

The purpose is to show that all examples follow the same finite DRCC count pattern.

For a finite problem instance P, the classical count is

$$N_{\mathrm{Classic}}(P) = |X(P)| \cdot c_{\mathrm{eval}}(P). \tag{9.1}$$

The DRCC count is decomposed as

$$N_{\mathrm{DRCC}}(P, \zeta) = N_{\mathrm{Collapse}}(P) + N_{\mathrm{Reconstruction}}(P, \zeta). \tag{9.2}$$

The runtime gain is

$$G(P, \zeta) = \frac{N_{\mathrm{Classic}}(P)}{N_{\mathrm{DRCC}}(P, \zeta)}. \tag{9.3}$$

Thus, DRCC is favorable under the specified count model only if

$$N_{\mathrm{Collapse}}(P) + N_{\mathrm{Reconstruction}}(P, \zeta) < N_{\mathrm{Classic}}(P). \tag{9.4}$$

Case	Classical Count	DRCC Count / Reduction	Condition for Gain
Housing	$n(c_{\mathrm{screen}} + c_{\mathrm{detail}})$	$nc_{\mathrm{screen}} + cc_{\mathrm{detail}}$	$c < n$, $c_{\mathrm{detail}} > 0$.
TSP / TSG	$(n-1)!/2$	$\leq n^2 + 2^{n-1} r_{\max}$	Controlled structural signatures and bounded reconstruction.
CSP	mq^n	$\leq mq^r + b(P)q^{\omega(P)+1}$	Small arity and bounded reconstruction width.
SAT / 3-SAT	$m2^n$	$\leq m2^r + b(\Phi)2^{\omega(\Phi)+1}$	Controlled clause width and reconstruction width.
Graph 3-Coloring	$m3^n$	$\leq 6m + b(G)3^{\omega(G)+1}$	Bounded coloring reconstruction width.
Full Adder	8	4 structural classes	Output depends on Hamming weight.

Across all cases, the same structural chain appears:

$$X(P) \longrightarrow Z(P) \longrightarrow \Omega_P(\zeta). \tag{9.5}$$

The classical model evaluates candidates in $X(P)$.

The DRCC model first constructs a reduced structural space $Z(P)$ and then reconstructs admissible candidates inside $\Omega_P(\zeta)$.

The transition condition is

$$G(P, \zeta) = 1. \tag{9.6}$$

Equivalently,

$$N_{\text{Classic}}(P) = N_{\text{DRCC}}(P, \zeta). \tag{9.7}$$

The count-level gap is

$$\Delta_N(P, \zeta) = N_{\text{Classic}}(P) - N_{\text{DRCC}}(P, \zeta). \tag{9.8}$$

A positive gap indicates that the DRCC count is smaller under the stated model.

A negative gap indicates that classical enumeration is smaller under the same model.

9.2 Non-Claim

The cross-case comparison does not prove that DRCC solves NP-complete problems in polynomial time.

It does not claim that SAT, CSP, graph coloring, or TSP become easy in the general worst-case sense.

It shows only that specified finite instances or structurally restricted families may admit useful reductions when reconstruction remains controlled.

$$\boxed{\text{DRCC is favorable only when controlled collapse and reconstruction together are cheaper than classical enumeration under the stated finite count model.}} \tag{9.9}$$

This is the common runtime conclusion of the case studies.

10 Literature Grounding and Final Position

10.1 Compact Literature Grounding

DRCC-V2 is positioned as a finite runtime-accounting and reconstruction framework.

It is related in spirit to several established directions, but it does not replace them.

Kernelization studies how problem instances can be reduced while preserving the relevant decision question.

Constraint propagation removes locally impossible assignments and makes implicit constraints explicit.

SAT solving methods such as DPLL, CDCL, resolution, and local search provide established algorithmic strategies for Boolean satisfiability.

Symmetry breaking avoids redundant exploration by identifying equivalent states.

Dynamic programming exploits reusable substructure in structured problems.

DRCC shares the general motivation of reducing redundant search, but its specific language is based on

$$d_{\text{frag}}(P) = |X(P)|, \tag{10.1}$$

$$d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|, \tag{10.2}$$

and

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta). \tag{10.3}$$

Thus, DRCC is not presented as a universal solver.

It is presented as a finite mathematical language for controlled collapse, admissible reconstruction, and count-level comparison.

10.2 Final Position

The central DRCC comparison is

$$N_{\text{Classic}}(P) \quad \text{versus} \quad N_{\text{DRCC}}(P, \zeta). \tag{10.4}$$

A DRCC reduction is favorable only when

$$N_{\text{DRCC}}(P, \zeta) < N_{\text{Classic}}(P) \tag{10.5}$$

under the specified candidate space, reduction map, reconstruction procedure, and cost model.

The manuscript does not claim a proof of $P = NP$, does not claim a proof of $P \neq NP$, and does not claim a universal speedup for SAT, CSP, graph coloring, routing, or any other problem class.

Its contribution is the following finite framework:

> DRCC studies when controlled structural reduction preserves enough information for admissible reconstruction while reducing the effective finite reconstruction space relative to classical enumeration. (10.6)

The case studies show how this principle can be expressed in concrete finite settings, including housing selection, TSP/TSG routing, CSP, SAT, graph 3-coloring, and the full adder.

10.3 References

[1] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Springer, 2013.

[2] R. Dechter, *Constraint Processing*, Morgan Kaufmann, 2003.

[3] M. Davis, G. Logemann, and D. Loveland, "A Machine Program for Theorem Proving," *Communications of the ACM*, 5(7), 394–397, 1962.

[4] J. Crawford, M. Ginsberg, E. Luks, and A. Roy, "Symmetry-Breaking Predicates for Search Problems," Proceedings of KR/AAAI, 1996.

11 Numerical Examples and Finite Case Evidence

11.1 Full Adder Example: Eight Inputs, Four Structural Classes

The full adder is the smallest complete numerical example in this manuscript.
It has three Boolean inputs:

$$a, b, c_{\text{in}} \in \{0, 1\}. \tag{11.1}$$

Therefore, the classical input space is

$$X_{\text{FA}} = \{0, 1\}^3. \tag{11.2}$$

Its size is

$$|X_{\text{FA}}| = 2^3 = 8. \tag{11.3}$$

The eight classical input states are

$$000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111. \tag{11.4}$$

Classically, one may treat these eight input states as eight separate cases.
DRCC uses the Hamming weight as the controlled reduction map:

$$C_{\text{FA}}(a, b, c_{\text{in}}) = a + b + c_{\text{in}}. \tag{11.5}$$

Thus,

$$C_{\text{FA}} : \{0, 1\}^3 \to \{0, 1, 2, 3\}. \tag{11.6}$$

The eight input states collapse into four structural classes:

$$C_0 = \{000\}, \tag{11.7}$$

$$C_1 = \{001, 010, 100\}, \tag{11.8}$$

$$C_2 = \{011, 101, 110\}, \tag{11.9}$$

and

$$C_3 = \{111\}. \tag{11.10}$$

Hence the finite DRCC collapse is

$$8 \longrightarrow 4. \tag{11.11}$$

The reconstruction dimensions of the four fibers are

$$d_{\text{rec}}(0) = 1, \qquad d_{\text{rec}}(1) = 3, \qquad d_{\text{rec}}(2) = 3, \qquad d_{\text{rec}}(3) = 1. \tag{11.12}$$

The full adder output depends only on the class value.
The sum bit is

$$s = C_{\mathrm{FA}}(a, b, c_{\mathrm{in}}) \bmod 2. \tag{11.13}$$

The carry-out bit is

$$c_{\mathrm{out}} = \begin{cases} 0, & C_{\mathrm{FA}}(a, b, c_{\mathrm{in}}) \in \{0, 1\}, \\ 1, & C_{\mathrm{FA}}(a, b, c_{\mathrm{in}}) \in \{2, 3\}. \end{cases} \tag{11.14}$$

The complete reconstruction table is:

Class	**Input States**	**Sum Bit**	**Carry-Out Bit**
C_0	000	0	0
C_1	001, 010, 100	1	0
C_2	011, 101, 110	0	1
C_3	111	1	1

The DRCC structural dimension is

$$R^{\mathrm{FA}}_{\mathrm{DRCC}} = 4. \tag{11.15}$$

The structural compression factor is

$$K^{\mathrm{FA}}_{\mathrm{DRCC}} = \frac{|X_{\mathrm{FA}}|}{R^{\mathrm{FA}}_{\mathrm{DRCC}}} = \frac{8}{4} = 2. \tag{11.16}$$

Runtime Comparison: Local and 16-Bit Full Adder

The local full adder has

$$d^{\mathrm{local}}_{\mathrm{frag}} = 8 \tag{11.17}$$

classical input states and

$$d^{\mathrm{local}}_{\mathrm{rec}} = 4 \tag{11.18}$$

Hamming-weight structural classes.
Thus, the local structural gain is

$$G_{\mathrm{local}} = \frac{8}{4} = 2. \tag{11.19}$$

For a 16-bit full adder, there are sixteen independent full-adder positions.
The classical local state space grows as

$$|T_{16}| = 8^{16} = 2^{48}. \tag{11.20}$$

Numerically,

$$|T_{16}| = 281{,}474{,}976{,}710{,}656 \approx 2.81 \cdot 10^{14}. \tag{11.21}$$

The DRCC structural class space grows as

$$|C_{16}| = 4^{16} = 2^{32}. \tag{11.22}$$

Numerically,

$$|C_{16}| = 4{,}294{,}967{,}296 \approx 4.29 \cdot 10^{9}. \tag{11.23}$$

Assume that the maximum reconstruction dimension per structural class is

$$d_{\mathrm{rec}}(C) = 3. \tag{11.24}$$

Then the DRCC count is

$$N_{\mathrm{DRCC}} = |C_{16}|\, d_{\mathrm{rec}}(C). \tag{11.25}$$

Thus,

$$N_{\mathrm{DRCC}} = 4{,}294{,}967{,}296 \cdot 3 = 12{,}884{,}901{,}888 \approx 1.29 \cdot 10^{10}. \tag{11.26}$$

The classical count is

$$N_{\mathrm{Classic}} = |T_{16}| = 2^{48}. \tag{11.27}$$

The resulting count-level gain is

$$G_{16} = \frac{N_{\mathrm{Classic}}}{N_{\mathrm{DRCC}}}. \tag{11.28}$$

Substituting the numerical values gives

$$G_{16} = \frac{281{,}474{,}976{,}710{,}656}{12{,}884{,}901{,}888} \approx 2.18 \cdot 10^{4}. \tag{11.29}$$

Equivalently, the 16-bit DRCC structural runtime model is about

$$21{,}800 \tag{11.30}$$

times smaller than the classical state enumeration model under the stated assumptions.
If one reference operation takes

$$f = 10^{9} \tag{11.31}$$

state updates per second, then

$$T_{\mathrm{Classic}} = \frac{2^{48}}{10^{9}} \approx 2.81 \cdot 10^{5} \text{ seconds}. \tag{11.32}$$

This is approximately

$$78.2 \text{ hours} \approx 3.26 \text{ days}. \tag{11.33}$$

For the DRCC count,

$$T_{\mathrm{DRCC}} = \frac{12{,}884{,}901{,}888}{10^{9}} \approx 12.9 \text{ seconds}. \tag{11.34}$$

Therefore, in this finite count-level model,

$$T_{\text{Classic}} \approx 78.2 \text{ hours} \qquad \text{versus} \qquad T_{\text{DRCC}} \approx 12.9 \text{ seconds.} \tag{11.35}$$

This runtime comparison is conditional on the structural DRCC model, the classwise reconstruction bound $d_{\text{rec}}(C) = 3$, and the reference update rate $f = 10^9$.
It is a count-level runtime comparison, not a hardware-level benchmark.

Quantity	**Value**	**Meaning**
Local classical states	8	One full-adder cell.
Local DRCC classes	4	Hamming weights $0, 1, 2, 3$.
16-bit classical states	$8^{16} = 2^{48}$	All classical local-state combinations.
16-bit DRCC classes	$4^{16} = 2^{32}$	All structural class combinations.
Max reconstruction size	$d_{\text{rec}}(C) = 3$	Largest local class size.
DRCC count	$4^{16} \cdot 3$	Class count times reconstruction bound.
Runtime gain	$\approx 2.18 \cdot 10^4$	Count-level structural gain.
Runtime at 10^9 Hz	78.2 h vs. 12.9 s	Classical versus DRCC count model.

11.2 Graph 3-Coloring Example: Four Vertices and Three Reconstructions

The graph 3-coloring example gives a finite reconstruction problem that can be checked by direct enumeration.
Consider the cycle graph

$$C_4 = (V, E) \tag{11.36}$$

with vertex set

$$V = \{v_1, v_2, v_3, v_4\} \tag{11.37}$$

and edge set

$$E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}\}. \tag{11.38}$$

Let the color set be

$$K = \{1, 2, 3\}. \tag{11.39}$$

A coloring is a map

$$c : V \to K. \tag{11.40}$$

The classical coloring space is

$$X_{\text{GC}}(C_4) = K^V. \tag{11.41}$$

Since there are four vertices and three colors, the classical candidate count is

$$|X_{\mathrm{GC}}(C_4)| = 3^4 = 81. \tag{11.42}$$

Thus,

$$d_{\mathrm{frag}}(C_4) = 81. \tag{11.43}$$

A coloring is admissible if adjacent vertices have different colors.
The local edge condition is

$$c(u) \neq c(v) \qquad \text{for every edge } \{u, v\} \in E. \tag{11.44}$$

Now fix the local fragment

$$c(v_1) = 1, \qquad c(v_2) = 2. \tag{11.45}$$

This fragment already satisfies the edge condition on $\{v_1, v_2\}$.
The remaining variables are

$$c(v_3), \qquad c(v_4). \tag{11.46}$$

They must satisfy the remaining edge constraints

$$c(v_3) \neq c(v_2), \tag{11.47}$$

$$c(v_4) \neq c(v_3), \tag{11.48}$$

and

$$c(v_4) \neq c(v_1). \tag{11.49}$$

Because $c(v_1) = 1$ and $c(v_2) = 2$, these conditions become

$$c(v_3) \neq 2, \tag{11.50}$$

$$c(v_4) \neq c(v_3), \tag{11.51}$$

and

$$c(v_4) \neq 1. \tag{11.52}$$

A direct finite check gives exactly three admissible completions:

$$(c(v_3), c(v_4)) \in \{(1, 2), (1, 3), (3, 2)\}. \tag{11.53}$$

Hence the reconstruction space is

$$\Omega_{\mathrm{GC}}(\zeta) = \{(1, 2, 1, 2), (1, 2, 1, 3), (1, 2, 3, 2)\}. \tag{11.54}$$

Therefore,

$$d_{\text{rec}}(C_4, \zeta) = |\Omega_{\text{GC}}(\zeta)| = 3. \tag{11.55}$$

The DRCC reduction for this fixed fragment is therefore

$$81 \longrightarrow 3. \tag{11.56}$$

The structural compression factor is

$$K_{\text{DRCC}}^{C_4} = \frac{d_{\text{frag}}(C_4)}{d_{\text{rec}}(C_4, \zeta)} = \frac{81}{3} = 27. \tag{11.57}$$

In the finite count model, the classical count is

$$N_{\text{Classic}}^{C_4} = 81. \tag{11.58}$$

The DRCC reconstruction count for the fixed fragment is

$$N_{\text{DRCC}}^{C_4} = 3. \tag{11.59}$$

Thus, the count-level gain is

$$G_{C_4} = \frac{N_{\text{Classic}}^{C_4}}{N_{\text{DRCC}}^{C_4}} = \frac{81}{3} = 27. \tag{11.60}$$

The result can be summarized as follows.

Quantity	**Value**	**Meaning**
Vertices	4	Cycle graph C_4.
Colors	3	Color set $K = \{1, 2, 3\}$.
Classical colorings	$3^4 = 81$	All possible color assignments.
Fixed fragment	$c(v_1) = 1,\ c(v_2) = 2$	Local DRCC fragment.
Admissible completions	3	Valid reconstructions after the fragment.
Compression factor	27	81/3.
Count-level gain	27	Classical count divided by reconstruction count.

This example is finite and directly verifiable.

It shows how a local graph-coloring fragment can reduce the effective reconstruction space from all colorings to a small admissible completion set.

It does not imply that arbitrary graph coloring becomes easy.

It shows only that this finite instance admits a controlled reconstruction after a specified local fragment.

11.3 Boolean Satisfiability Example

Consider a small 3-SAT instance with three variables. The classical assignment space is

$$X_{\text{SAT}} = \{0, 1\}^3. \tag{11.61}$$

Therefore,

$$|X_{\mathrm{SAT}}| = 2^3 = 8. \tag{11.62}$$

Let the clause-output map group assignments by the vector of satisfied clauses:

$$\sigma_\Phi(a) = (K_1(a), K_2(a), K_3(a)). \tag{11.63}$$

Assume that the eight assignments produce four distinct structural clause-output classes. Then

$$8 \longrightarrow 4. \tag{11.64}$$

The DRCC structural rank is

$$R_{\mathrm{DRCC}}(\Phi) = 4. \tag{11.65}$$

The finite compression factor is

$$K_{\mathrm{DRCC}}^{\mathrm{SAT}} = \frac{8}{4} = 2. \tag{11.66}$$

If the satisfying class is represented by

$$(1, 1, 1), \tag{11.67}$$

then the reconstruction space is

$$\Omega_{\mathrm{SAT}}(1, 1, 1) = \{a \in \{0, 1\}^3 : \sigma_\Phi(a) = (1, 1, 1)\}. \tag{11.68}$$

The reconstruction dimension is

$$d_{\mathrm{rec}}(\Phi) = |\Omega_{\mathrm{SAT}}(1, 1, 1)|. \tag{11.69}$$

This example shows how SAT assignments may be grouped by clause-output behavior. It does not claim a general SAT solver.

11.4 Constraint Satisfaction Example

Consider a CSP with

$$n = 4, \qquad q = 3. \tag{11.70}$$

The classical assignment space has size

$$q^n = 3^4 = 81. \tag{11.71}$$

Assume that local constraint tables reduce the search to a reconstruction space with

$$d_{\mathrm{rec}} = 9. \tag{11.72}$$

Then the finite DRCC reduction is

$$81 \longrightarrow 9. \tag{11.73}$$

The compression factor is

$$K_{\mathrm{DRCC}}^{\mathrm{CSP}} = \frac{81}{9} = 9. \tag{11.74}$$

In count-level form,

$$N_{\mathrm{Classic}} = 81, \qquad N_{\mathrm{DRCC}} = 9. \tag{11.75}$$

Thus,

$$G_{\mathrm{CSP}} = \frac{81}{9} = 9. \tag{11.76}$$

This example is conditional on the stated local reduction and reconstruction space. It does not claim that all CSP instances admit such a reduction.

11.5 TSG / TSP Routing Example

Consider a symmetric travelling-salesman-type routing problem with

$$n = 6 \tag{11.77}$$

cities and one fixed starting city.
The number of classical tours is

$$N_{\mathrm{Classic}} = \frac{(6-1)!}{2} = \frac{120}{2} = 60. \tag{11.78}$$

Assume that a DRCC structural route reduction produces

$$18 \tag{11.79}$$

route classes, and that at most

$$r_{\max} = 3 \tag{11.80}$$

representatives are reconstructed per class.
Then the DRCC count is bounded by

$$N_{\mathrm{DRCC}} = 18 \cdot 3 = 54. \tag{11.81}$$

The count-level gain is

$$G_{\mathrm{TSP}} = \frac{60}{54} \approx 1.11. \tag{11.82}$$

This small instance shows only a weak gain, because $n = 6$ is still small. The example illustrates that DRCC overhead and reconstruction budget matter. It does not claim a general TSP solution.

11.6 Housing Example

Consider a housing selection instance with

$$n = 1200 \tag{11.83}$$

available apartments and

$$c = 25 \tag{11.84}$$

remaining candidates after structural filtering.
Let

$$c_{\text{screen}} = 7 \tag{11.85}$$

and

$$c_{\text{detail}} = 1000. \tag{11.86}$$

The classical count is

$$N_{\text{Classic}} = 1200(7 + 1000) = 1{,}208{,}400. \tag{11.87}$$

The DRCC count is

$$N_{\text{DRCC}} = 1200 \cdot 7 + 25 \cdot 1000 = 33{,}400. \tag{11.88}$$

Thus,

$$G_{\text{Housing}} = \frac{1{,}208{,}400}{33{,}400} \approx 36.18. \tag{11.89}$$

The gain appears because detailed inspection is applied only to the reduced reconstruction set. This is not a universal filtering speedup.

11.7 Cross-Case Numerical Summary

The finite examples can be summarized by the same count-level pattern:

$$G = \frac{N_{\text{Classic}}}{N_{\text{DRCC}}}. \tag{11.90}$$

Example	**Classical Count**	**DRCC Count**	**Gain**
Full Adder	8	4	2
Graph C_4 Coloring	81	3	27
Small 3-SAT	8	4	2
Small CSP	81	9	9
Six-City TSP	60	54	≈ 1.11
Housing	1,208,400	33,400	≈ 36.18
16-Bit Full Adder	2^{48}	$3 \cdot 2^{32}$	$\approx 2.18 \cdot 10^4$

The table shows that DRCC is not one universal algorithm. It is a finite reconstruction framework.

A gain appears only when the controlled reduction and reconstruction count is smaller than the classical count:

$$N_{\text{DRCC}} < N_{\text{Classic}}. \tag{11.91}$$

This is the common numerical conclusion of the examples.

11.8 Housing Problem: Reduced Candidate Inspection

Consider a housing selection instance with

$$n = 1200 \tag{11.92}$$

available apartments and

$$c = 25 \tag{11.93}$$

remaining candidates after structural filtering.
Let

$$c_{\text{screen}} = 7 \tag{11.94}$$

and

$$c_{\text{detail}} = 1000. \tag{11.95}$$

The classical count is

$$N_{\text{Classic}} = 1200(7 + 1000) = 1{,}208{,}400. \tag{11.96}$$

The DRCC count is

$$N_{\text{DRCC}} = 1200 \cdot 7 + 25 \cdot 1000 = 33{,}400. \tag{11.97}$$

Thus,

$$G_{\text{Housing}} = \frac{1{,}208{,}400}{33{,}400} \approx 36.18. \tag{11.98}$$

This gain appears because detailed inspection is applied only to the reduced reconstruction set.
The example does not claim a universal speedup for filtering tasks. It shows only that controlled reduction can be useful when detailed evaluation is expensive and the reduced candidate set remains small.

11.9 Final Numerical Conclusion

The numerical examples in this chapter show the same finite DRCC pattern.
Classical enumeration starts from a candidate space

$$X(P). \tag{11.99}$$

DRCC replaces direct enumeration by the reconstruction chain

$$X(P) \longrightarrow Z(P) \longrightarrow \Omega_P(\zeta). \tag{11.100}$$

The common comparison is

$$N_{\text{Classic}}(P) \quad \text{versus} \quad N_{\text{DRCC}}(P, \zeta). \tag{11.101}$$

A numerical DRCC advantage is present only when

$$N_{\mathrm{DRCC}}(P,\zeta) < N_{\mathrm{Classic}}(P). \tag{11.102}$$

The examples do not establish a universal algorithmic speedup and do not resolve the P versus NP problem.

They show that, in finite and explicitly specified settings, controlled structural reduction can reduce the effective reconstruction space while preserving the admissible output or solution structure.

$$\boxed{\text{DRCC is a finite reconstruction framework: it is useful when structural collapse preserves admissible reconstruction and the resulting reconstruction count is smaller than classical enumeration.}} \tag{11.103}$$

This example is finite, complete, and directly verifiable.

It shows that the full adder output can be reconstructed from the Hamming-weight class rather than from the identity of the individual raw input state.

The reduction is output-preserving but not input-invertible.

It does not imply that arbitrary Boolean circuits admit the same collapse.

Appendix A DRCC Formula and Boundary Sheet

12.1 Core DRCC Quantities

Symbol	**Meaning**
$X(P)$	Classical candidate space of the finite problem instance P.
$X_{\mathrm{adm}}(P)$	Admissible candidates satisfying the constraints of P.
$C_P : X(P) \to Z(P)$	Controlled DRCC reduction map.
$Z(P)$	Reduced structural state space.
$\zeta \in Z(P)$	Reduced structural state or class.
$\Omega_P(\zeta)$	Reconstruction space associated with ζ.
$d_{\mathrm{frag}}(P)$	Fragmentation dimension.
$d_{\mathrm{rec}}(P,\zeta)$	Reconstruction dimension.
$N_{\mathrm{Classic}}(P)$	Classical evaluation count.
$N_{\mathrm{DRCC}}(P,\zeta)$	Total DRCC evaluation count.
$G(P,\zeta)$	Count-level runtime gain.

12.2 Core Formula Sheet

The fragmentation dimension is

$$d_{\mathrm{frag}}(P) = |X(P)|. \tag{12.1}$$

The reconstruction space is

$$\Omega_P(\zeta) = \{x \in X_{\mathrm{adm}}(P) : C_P(x) = \zeta\}. \tag{12.2}$$

The reconstruction dimension is

$$d_{\mathrm{rec}}(P,\zeta) = |\Omega_P(\zeta)|. \tag{12.3}$$

The DRCC structural stability condition is

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty. \tag{12.4}$$

The DRCC count decomposition is

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta). \tag{12.5}$$

The count-level runtime gain is

$$G(P, \zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P, \zeta)}. \tag{12.6}$$

A DRCC advantage is present only if

$$N_{\text{DRCC}}(P, \zeta) < N_{\text{Classic}}(P). \tag{12.7}$$

12.3 Rules of Acceptance

Rule	**Meaning**
RB1	A controlled structural representation exists.
RB2	The reduced structural state space is constructible.
RB3	Admissible reconstruction is preserved under the reduction.
RB4	The reconstruction count or state dimension remains controlled under the specified model.

12.4 Boundary Statement

DRCC does not claim a universal runtime speedup.

It does not claim a proof of $P = NP$ or $P \neq NP$.

It does not claim that SAT, CSP, graph coloring, TSP, or routing problems become easy in the general worst-case sense.

The manuscript supports the following finite statement:

$$\boxed{\text{DRCC is useful when controlled structural reduction preserves admissible reconstruction and the resulting reconstruction count is smaller than classical enumeration under the stated finite count model.}} \tag{12.8}$$

12.5 Final Closing Statement

The present manuscript formulates DRCC as a finite reconstruction framework.

Its central purpose is to compare classical enumeration with controlled structural reduction and admissible reconstruction.

All numerical examples are interpreted as finite count-level evidence under explicitly stated assumptions.

No universal speedup theorem is claimed.

No conclusion about $P = NP$ or $P \neq NP$ is asserted.

The contribution of DRCC is a structured language for identifying when collapse, reconstruction, and runtime accounting can be made explicit.